PERSONALNI
računari
97
JANUAR
1994.
SLT 500 • Denara 4000 • DEM 10 • USD 6 • ATS 70

SREĆNA NOVA GODINA

POL KRNETA:
ŠTA SAM
PREĆUTAO O ALFI

USENET

CHICAGO
DOS 6.2

HP LASERJET 4P

PRVE PENTIUM MAŠINE

SOFTVER

VISUAL BASIC 3.0

QUATTRO PRO
FOR WINDOWS

SPECIJALNI DODATAK NA 32 STRANE

CLIPPER 5.2

KOMPLETNO UPUTSTVO ZA UPOTREBU

Promenili smo brojeve telefona !

"OSA" RAČUNARSKI INŽENJERING, NARODNOG FRONTA 56, BEOGRAD, TEL. 681-178, 681-199, FAX. 681-743

PERSONALNI
računari 97

časopis za informatiku i računarstvo
YU ISSN 0352-7221
Izdaje BIGZ
Januar 1994

Izdaje i štampa
Beogradski izdavačko-grafički zavod
11000 Beograd
Bulevar Vojvode Mišića 17
Generalni direktor
Ilija Rapajić
Glavni i odgovorni urednik
Jovan Regasek
Zamenik glavnog i odgovornog urednika
Ranka Jovanović
Komercijalni urednik
Vesna Jeremić
Tehnički urednik
Radoje Kavedžić
Stručna redakcija
Nenad Ratković (baze podataka) Ranka Jovanović (aktuelnosti) Dejan Ristanović (softver) Jovan Skuljan (programiranje) Vladimir Stamenović (radne tabele) Vesna Ćorić (aktuelnosti) Zoran Živolić (softver)
Adresa redakcije
11000 Beograd
Bulevar vojvode Mišića 17/III
Telefoni
647-476 (glavni urednik)
653-748 (redakcija)
651-666 (centrala preplata 226, stan Ivanjevi 256)
Agencija BIGZ-a
(011) 651 793 653-586
Telefaks
(011) 648-140 647 956
Pretplata za zemlju

Za 3 meseca	zvati prodaju
Za 6 meseci	zvati prodaju
Za 11 meseci	zvati prodaju

Ina žiro-račun D P BIGZ, 60802-603-1-23264)
Pretplata za inostranstvo
Za jednu godinu 51 USD 80 DEM 67 CHS 27 GBP 271 FRF
(na devizni račun D P BIGZ 60811-620-16101 620701 566-03377)
Zbog usporenog prenošenja uplata preko banke molimo pretplatnike da nam pošte svake nove uplate odmah pošalju foto-kopiju uplatnice
Rukopisi se ne vraćaju

SEZAM (011) 648-422 (15 linija)
sistem za mašinske komunikacije
Upravnik sistema
Jovan Regasek
Administratori na sistemu
Zoran Živolić i Dejan Ristanović

CENOVNIK OGLASNOG PROSTORA

Cene oglasnog prostora se izražavaju u bodovima čija vrednost odgovara vrednosti nemačke marke na slobodnom tržištu na dan fakturisanja

- 1/1 poslednja strana korica 1.000 bodova
- 1/1 druga strana korica 900 bodova
- 1/1 treca strana korica 900 bodova
- 1/1 kolor unutrašnja strana 800 bodova
- 1/2 kolor unutrašnja strana 500 bodova
- 1/1 crno-bela strana 550 bodova
- 1/2 crno-bela strana 400 bodova
- 1/4 crno-bela strana 300 bodova
- 1/8 crno-bela strana 200 bodova
- mali oglas (po centimetru) 20 bodova

Fakturisanje se vrši na dan ugovaranja oglasnog prostora i prijema potrebnih materijala za oglas, sa obavezom da se uplata izvrši pre izlaska broja u štampu

Rok za dogovor i rezervaciju oglasnog prostora je 35 dana pre izlaska broja iz štampe

Molimo vas da se za dogovore i informacije javite na telefon 011/653-748 fax 011/648-140 modem 011/648-422 (mail write redakcija) Kontakt: **Vesna Jeremić**

SADRŽAJ

4 Vesti
ŠTA IMA NOVO

7 Intervju / Pol Kmeta
ŠTA VAM NISAM REKAO

10 Nagradna igra
VELIKO FINALE

12 Mikroprocesori / Pentium
PRVE LASTE

20 Laserski štampači / HP 4P
ZA LIČNU UPOTREBU

22 Operativni sistemi / Windows 4.0
CHICAGO BEZ MASKE

29 Operativni sistemi / DOS 6.2
IZGRADITE SVOJ DOS SVET

32 Radne tabele / QPW 5.0
NIJE LAKO O GAOJEVIMA

36 Mikroprocesori / AMD486 40 MHz
MOŽE I BOLJE

37 Mikroprocesori / Cx486DRX^2
BRŽI I VOZI

38 Programski jezici / Visual Basic 3.0
SAVRŠENA JEDNOSTAVNOST

42 Alatke za pisanje / Grammatik 5.0
GRAMATIKA NIJE GAUK

45 Firme
PC PLUS, SAGA

48 Vesti
ŠTA IMA NOVO

52 Dejanove pitalice
MUKE STARIH EGIPĆANA

54 Komunikacije / Usenet
PRVI OGLASNIK SVETA

58 Tehnike programiranja / Detekcija grešaka
IZUZETNO DOBRA STVAR

62 Protokoli / DPMI
PROŠIRENI DOS

63 Ne prvi pogled / Rasterex
LAKI POGLED IZ RASTEREX-a

64 Tehnike programiranja / Modemi
ČEKAJUĆI NA VEZU

66 Tehnike programiranja / Paradox Engine
OBJEKTI, TABELE, INDEKSI

69 Zanimljivosti / Virusi
DEDETI KOMENDAN VIRUSA

71 Zoran Kohler
WIN.INI

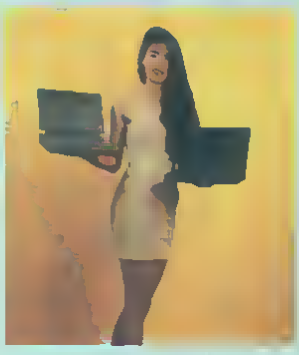
72 Pavle Peković
MOJA ŠASLA UNIX-a

73 Zoran Živolić
MOJA ŠKOLA C-a

74 Dejan Ristanović
NAJTEŽI LIČNE PRIRODE

76 Modemske komunikacije / Sezam
CODE NAME SEZAM

61 Bilteni
SEZAM FILE



Žana Ranković, manekenka iz Kosljerića, pokušava da odvagne za koju elektronsku belešnicu da se odluči lako vitka i krhka, ona ipak o laptopu drži na dlanovima dva računara, kao da pripadaju *palmtop* a ne *notebook* kategoriji. Koji je od njih lakši, nije teško ustanoviti, ali sama „lakoća" nije presudna za njihovu ocenu. Bo sme biti u prilici da saznamo šta se stvarno krije u ovim malim, elegantnim kutijama – oba modela su se našla u grupi od nekoliko notebook računara čije testiranje privodimo kraju. Žana, naravno, već zna o čemu se radi, a vi potražite rezultate testiranja u našem sledećem broju. Primer je ovekovečen u Studiju Tošić.

SADRŽAJ OGLASA

ADA COM	9, 35
ASYS COMERCE	37,46,48
BIOSFERA	17
BIPIP	53
BS PROCESOR	28
COMPUTER BEOGRAD	51, 70
COMTRAD	4K, 18
ELEGRA	41
F&D LAB	53
IMTEL	50
INSTUT M KIDRIČ	82
INTERSOFT	27
JITEX	31
JUGODATA	5
MICRO AIR	53
MICROSYS	18,19
MR SYSTEM	46
MZ COMPUTER	44
OLIVETTI	30
OLYMP	47
OSA	2K
PERIHARD	51
PROSOFT	66
PS PLUS	6
RIBON	70
SAGA	3K
SPRINT	60
ŠUTLIĆ	61

„Računari" se od ovog broja više neće prodavati u prodajnoj mreži „Politike" već samo u knjižarama BIGZ-a, se uobičajenim popustom od 20% Ukoliko u vašem mestu nema BIGZ-ove knjižare, svoj primerak možete naručiti preko naše službe za prodaju i pretplatu (tel 651-666, lokal 226) Kompletan spisak prodajnih mesta i uslove za pretplatu potražite u specijalnom dodatku u ovom broju „Računari 93"

SOFTVER

Razvojno okruženje

Visual C++: Korak ka boljem i lakšem povezivanju

Devetog decembra je na velikoj predstavi koju je preko satelita pratilo više od 30 hiljada ljudi u Americi i Evropi predstavljen novi razvojni sistem *Microsoft Visual C++ 1.5*. Najveće promene u odnosu na verziju 1.0 su nove mogućnosti „čarobnjaka" (razvojnih alata koje je Microsoft nazvao *wizards*) i značajno poboljšanje in dustrijskog standarda MFC (*Microsoft Foundation Class*)

Novi MFC 2.5 omogućava lakšu kontrolu nad OLE-om 2.0 i ODBC standardom za povezivanje se bazama podataka. Tskoda što nije manje značajno iseraestobitne aplikacije za DOS i *Windows 3.1* sada možete razvijati i pod operativnim sistemom *Windows NT*

Visual C++ 1.5 na tržištu je zamenio *Visual C++ 1.0 Professional Edition* što znači da se radi o punom paketu koji zauzima izuzetno veliki prostor. Stoga se proizvođač odlučio da ga ponudi i u verziji na CD-ROM disku. Tako se na jednom mestu nalazi 150 megabajta programa i podataka, sa kompletnom dokumentacijom.

U okviru razvojnog paketa nalazi se i OLE 2.0 kompatibilan se svim aplikacijama iz paketa *Microsoft Office* kao i *Visual Basic 3.0*. Ovako složen softverski sistem bio bi daleko teži za korišćenje u razvoju novih aplikacija da nema standarda *Microsoft Foundation Class* MFC tool, dug oko 20 hiljada linija napisanih C++ programskim jezikom pokriva celokupni spektar OLE 2.0 mogućnosti: vizuelno editovanje OLE 2.0 drag-and-drop OLE automatizaciju strukturiranog skladištenja (*structured storage*) i drugo

MFC 2.5 takođe omogućava i lako povezivanje ODBC standarda se aplikacijama koje se razvijaju. Ta tic programi mogu da pristupaju raznolikim formatima baza podataka, kao što su *Access*, *dBASE*, *Paradox*, *Sirrvos SQL Server* i *Oracle*. MFC 2.5 podrške bazama podataka sadrži celokupni ODBC API kombinovan sa C++ klasama i apstrakcijama se pristup podacima.

Visual C++ 1.5 omogućava da treinate programe se DOS i *Windows 3.1* na računarima koji rade pod *Windows NT* operativnim sistemom što će kako predviđaju u Microsoftu uskoro biti najčešća platforma se razvoj aplikacija.

Da bista koristili *Visual C++ 1.5* potreban vam je računar se 80386 ili jačim procesorom 8 megabajta o perativne memorije CD-ROM drajv od 8 do 80 MB prostora na disku najmanja VGA video podsistem DOS 5.0 ili noviji i naravno, *Windows 3.1*, odnosno *Windows NT*

DOMAĆE TRŽIŠTE

Jubileji

DSA – Prvo četiri godine

U beogradskom hotelu „Hyatt Regency" 24 novembra ove godine kompanija „DSA – računarski inženjering" je prezentacijom svoje dalatnosti obeležila četiri godine postojanja Delatnost OSE je vrlo bogata – pored prodaje softvera i hardvera projektovanja sistema, lokalne aplikacije i računarskog inžinjeringa, OSA se u poslednje vreme sve više bavi multimedijom vizuelizacijom i animacijom

Iz hardverska i softverska ponuda izdvojani su i i zuzetno efektno predstavljeni proizvodi Autodesk a Rasterax-a i Oce Graphics-a sti i nekoliko sopstvenih projekta – kompletan informacioni sistem aerodroma u Kemarovu (Rusija) kao i sistem za arhiviranje tehničke dokumentacije u Rafinerji Novi Sad

Visoko-profesionalni način na koji se OSA ovom prilikom predstavila najbolje od svega govori da se radi o firmi koja na ove prostore uvodi najviše standarde i najsavremenija trendove u primeni računara u dokumentaristici, arhitekturi i medijima.

DAJ BOŽE

Pre ravno jedne decenije negde upravo ovih dana, uoči samo Nove 1984 godine na kioscima širom tadašnje Jugoslavije osvanuo je prvi broj časopisa „Računari". On je stvoren u redakciji tada veoma ugledne ško la za popularizaciju nauke „Galaksija" a izašao je kao specijalno izdanje pod naslovom „Računari u vašoj kući" Pokazaće se da je taj gotovo proročanski naslov ohrabrio da je definitivno kucnuo čas da računari udu u – naše živote U tom trenutku ipak nika nije mogao da zna da li će specijalno izdanje preraziti u časopis ili ne. Na jednoj od promocija lista u prepunoj knjižari u Novom Sadu prisutnikrao sam razgovor dva mladića „Kažu da će Računari izaći i dalje" rekao je jedan „Daj Bože!" uskliknuo je drugi skloplenih šaka i pogleda uprtog ka nebu Iz tog „Daj Bože" redakcija i danas crpi energiju za svaki novi broj

U svetu ima veoma mnogo računarskih časopisa. Njihov broj se meri desetinama pa i stotinama. Po pravilu što je zemlja tehnološki razvijenija ime i više računarske štampe To je zato što su računarski časopisi veoma važan pratilac mogao bih da kažem i sastavnik računarske industrije U pitanju je veoma živa, moćna najdinamičnije industrijska grana, u kojoj se tehnologije smenjuju svakih godinu, dve ili tri Računarski časopis pre svega informišu o novim tehnologijama, s jedne strane a s druge obrazuju za njihovu primenu I tako paspešuju njihov razvoj Ne mogu da zamislim kako bi se razvijala čitava stvar bez tog stravičnog protoka informacija koji se odvija kroz računarske časopise

„Računari" su prvi najstariji domaći kompjuterski časopis a sleriji su I od mnogih sličnih listova u svetu Nastojeći da pre svega budu i ostanu „tehnološke preihodnica" „Računari" su često bili više od običnog časopisa Mi nismo samo pratili ono što se dešava, nego smo i brojnim akcijama najdirektnije učestvovali u stvaranju onoga što se zove „domaće tržište" Sa prvim brojem lista smo lansirali o samogradnji mali računar „galaksija" (u vreme kada je uvoz kompjutera bio praktično zabranjen) koji je gotova čitava jednu generaciju uveo u čudesni svet računara Nešto kasnije kada su postali interesantni PC računari a granice za tu robu o pet bile zatvorene mi smo kroz akcije „Minhenske" i „Singapurska veza" pronalazili načine za njihov masovni uvoz. Pre četiri godine da nabrojimo samo ono naj važnije osnovali smo Sistem se moderneke komunikacije Sezam koji danas okuplja nekoliko hiljada članova i stoji rame uz rame se najvećim BBS mrežame u Evropi

Deset godina je dugačak period u i ljudskom životu, a u istoriji jednog časopisa pogotovu I iet je sa promenama u tehnologiji koju je pratio, i sem često manjeo svoj lik Ponekad I uz velike otpore čitalaca. Sećam sa da su prvi tekstovi o višestruko skupljim PC računarima, u vreme kada je čitav hakarski svet govorio sam o „spektrumima" i „komodorima" primljeni prilično hladno Gotovo na nož. Mi smo ipak relativno rano još dok su oni bili ne vrhunou popularnosti razkrstili se tim izv „malim kućnim računarima" video igrama i piratskim oglasima dekle se kompjuterima kao igračkama I okrenuli se PC računarima koji se koriste prvenstveno kao alatka za obavljanja određenih poslova

Taj oštri koncepcijski zaokret smo načinili se izvesnom zebnjom u srcu Sada znamo da su „Računari" u tom času zapravo posisli punoletni, i srećni smo što se to dogodilo na vreme Čitaoci su dobili kompetentan list, a domaće ponude direktan i veome efikasan kanal do svojih kupaca. Oslonac na tržište na kome list i danas stuži kao svojevrstan katalizator između onih koji prodaju računarsku opremu i onih kojima je ta oprema potrebna, brzo nam je omogućio da sve elemente časopisa – izbor tema, način njihove obrade i grafički izgled – usklađimo se svetskim standardima u ovoj vrsti publicistike.

Tih visokih standarda se neka nam huda dozvoljeno malo slavljeničke neskromnosti držimo i danas – uprkos svemu. Znamo da je cena lista i cene naših usluge određenim kategorijama čitalaca i poslovnih partnera sve nedostupnija i zbog toga duodko žalimo Ali kao što računarska industrija nikada neće odustati od 486 mašina da bi ponovo proizvodile XT računare ni mi ne vidimo ni razloga ni načina da se vratimo unazad

„Računara" što je osavim uobličajeno se ovu vrstu časopisa, stvara mala redakcija i male akipe spoljnih saradnika ali se, verujte još nismo umorili Naprotiv tek smo sad u prevoj formi za – sledeću decenju Za sve ove godine kroz redakciju je prošlo dosta autora čiji su prilozi ostavili i ostavljaju duodk trag – u istorijskom nasleđu Voja Antonić Andelko Zgonjec Nevenka Spalević Jelena Rupnik Vladimir Kostić Jovan Skuljan Duško Savić prof dr Dušan Slavić Dragutin Vuković Vesna Ćosić Dejan Veselinović Nenad Batočanin Vladimir Stamenović Bojan Petrović Dejan Jelović Zoran Kehler – ali tekstove u osnovi piše ieto autorsko jezgo od prvog broja U srcu tog jezgra nalaze se Dejan Ristanović i Zoran Životić, bez čijh se priloga ne može zamisliti ni jeden broj Osećam ispisujući ova pomalo svečarske redove potrebu da se zahvalim svim našim autorima, bez obzira da li objavili jeden ili stotinu tekstova, na onoma što su ugradili u list.

Zbog nezdrživih uslova poslovanja, „Računari" se za izvesno vreme od ovog broja potpuno povlače sa kioska. Trenutno se mogu kupiti samo u lancu IKGZ ovih knjižara, koji srećom, dobro pokriva tržište, ili putem preplate Tiraž lista je bez odzira na teškoće sa plasmanom i dalje stabilan se nekoliko prijatnih skokove u toku godine – čitaoci kojima je list potreban koji svaki broj očekuju se „daj Bože" uvek će naći načina da do njega dođu Dok ja njih – biće i „Računara"

Jovan Regasek

Ovo su „Računari"

	Ukupne tekst		Ukupne strana	
Programerski tekstovi	901	(34%)	2241 75	(36%)
Korisnički tekstovi	562	(21%)	1572 75	(25%)
Tržišni prikazi	373	(14%)	818 50	(13%)
Reklamni tekstovi	183	(7%)	386.50	(6%)
Hardverski tekstovi	143	(5%)	313.00	(5%)
Računari u obrazovanju	80	(3%)	148.25	(3%)
Ostali tekstovi	411	(16%)	719.25	(12%)

Desetogodišnja statistika. U bazi RtND na Sezamu trenutno se nalazi 2663 teksteva, računajući tu i i zdanje „Računari u vašoj škeli" Ukupna dužina tih tekstova (izražena u stranicama „Računara", ne uzimajući u obzir da je kod promene formata i sloga bila promena broja karaktera na strani) iznosi 6207 Indeks ne obuhvata vesti.

Top lista autora svih vremena

Autor	Teksteva
Dejan Ristanović	536
Zoran Živetić	154
Dejan Veselinović	112
Žarko Berberski	95
Nevenka Spalević	74
Jovan Skuljan	57
Dušku Savić	53
Marko Krić	45
Nenad Batočanin	43
Vlada Kostić	35
Vesna Ćastć	31

Prvih jedanaest autora prema broju objavljih priloga. Lista autora više od deset objavljanih priloga objavljivanja na strani 98, u akviru registra tema se 1993. godinu

Tri želje za petogodišnjicu

Želeo bih
da kupim najbolji sistem za moj biznis!

Želeo bih
da za moj biznis obezbedim
sav neophodan software za lakši rad!

Želeo bih
da budem siguran da sam kupio sistem od
firme koja može da mi pruži odličan servis i održavanje!

1994.
jugodata

...dućnost Vaše ... na nama!

jugodata
RAČUNARSKI SISTEMI

B. Revolucije 326, 1[illegible]0 Beograd; tel/fax: (011) 41[illegible]6, 4[illegible]9-768, (081) 12-403

PC PLUS, Budimska 10 a, Beograd, (011) 337-120, 340-22

ŠTA VAM NISAM REKAO

Kada je pre godinu dana procurilo da će Alfa raditi na 200 MHz, gospodin Scott McNealy, glavni izvršni direktor Sun-a, je rekao: „Only over my dead body!" Nakon objavljivanja Alfe, DEC je, sasvim anegdotski, u nizu časopisa puetio sledeću reklamnu poruku: „Scott, it's your turn now!" Zbog kontroverzi, (između ostalog) sasvim anegdotskih, koje je u domaćem okruženju pokrenuo Intervju sa „alfacentričnim" Polom Krnetom, dele sam sebi trude da, ekskluzivno za Računare, zabeležim novi razgovor sa mojim „dekocentričnim" prijateljem. 19. decembra, na Svetog Nikolu, gospodin Pol Krneta, između Škotska i Kalifornije, ponovo me je totalno zaveo ALFA računarima i okružio ALFA litereturom.

Vesna Ćosić

Računari: Šta je to što u arhitekturi Alfa kreira te izuzetne rezultate?

Pol Krneta: Arhitektura je ono što čini mašinu, ostaje večito ukucano u nju što ne može da se promeni (registri način obrade). Trenutna implementacija je takva da Alfa ima taj mnogo manji limit od svega nekoliko gigabajta što je sveže čekirt puta više od toga. Za one kojima je to malo od ključne podine ove nove generacije čipova koji imaju četrdeset adresnih linija, tako da će moći da adresiraju milijardu gigabajta fizičke memorije. Ko sa želi na 34 bita, može preći i tih 64 gigabajta memorije.

Računari: Koliko bi to koštalo?

Pol Krneta: Oko sto hiljada dolara po gigabajtu. Mislim to čini oko milion i po dolara (smeje se). Ja ne znam. Znam cene nekih drugih memorija. I ne tamu da li je Alfa sada 64-bitna, zato što ima adresna bila potpuno je akademska. Novi čip sa sto hiljadu gigabajta fizičke memorije. Ista priča kao sa Intelom 386 i 386SL. 386 ima 32-bitni adresni i 32-bitni *data bus*. SL ima 24-bitni adresni bus čip je i dalje 32-bitni jedino što ne može da adresira punih 4Gb nego adresire samo 16Mb. Stvar je u implementacije ništa drugo. Tako da je za sada Alfa ograničena na mizernih 16Gb. Neva je ne 16Gb. Poenta je u tome što su ova high-end mašine Alpha 4000 i Alpha 7000 otvorene samo po sebi. U ovom trenutku one imaju četrdeset adresnih bita. Čovek može u njih da spakuje 1000Gb memorije. DEC je dao kompletnu dokumentaciju za te *high-end* mašine ko misli da može da napravi bolju mašinu neka je napravi. Recimo u Alfi ima 7 mesta za memorijske kartice. Te kartice su sada najveće kod DEC-a – 2Gb po kartici. Ko misli da može da napravi bolje, dobija pune specifikacije. Ako misli da može da napravi mašinu sa 16Gb po kartici onda to to bilo 112 GB u mašini. Neka samo izvoli! Stvar je vrlo otvorena. Bilo je sve stare vremena kada je DEC otvorio U-BUS i Q-BUS kada je svako pravio kartice za njega. Ako neko ima znanje može da napravi bukvalno sve što hoće. Kao što su u firmi Avalon napravili Alfa kompjuter za VAX-a. A oni su samo jedna od 150 firmi koje kupuju Alfa čipove i pravi svoje računare. Carnera Multiformije sada pravi Alfa PC ja. Kupuje od DEC-a čipove i pravi. To jeste konkurencija DEC-u ali i pokazatelj da je u pitanju zdrave stvar. Čovek može da kaže – neću od DEC-a kupiću od svih sa *Windows*-om ili sa OSF-am. To je to što je lepo u celoj priči – otvorena stvari.

Računari: A konverzija high-end softvera na 64-bitnu arhitekturu?

Pol Krneta: U Školskoj je DEC-ova fabrika računara. Mi smo se jednom vodećom svetskom firmom videze podataka ušli u ispitivanja. Pre toga smo u okviru pogledali njihov *source* da vidimo da li je performansan za Alfu. Tu smo našli da performanse su bili znatno bolje oni kopiraju na primer bajt umesto osam bajtova odjednom što daje znatno ubrzanje. Usput smo došli na ideju da probamo na tj njihov server tj baza podataka. Iz cuga buda Informix. Pola dana smo se sa tim igrali. Zamanili smo nas *short* inkiščere koji su tipični za *baby-unix* sa *long* inkiščerima. Posle pola dana je krenulo strašno jako je teško testirati mašinu koje može da ima 3.5Gb fizičke memorije u jednoprocesorskoj konfiguraciji. Takvih mašine nema puno po ovom svetu osim onih van DEC-a. Zato smo krajem novembra ili početkom jednu takvu mašinu u Školskoj. Okupilo se tim prilikom jedno petnaestak ljudi, iz celog sveta, da vide kako se ponaša ta baza podataka na mašini koje je po koncepciji izvan domena *mainframe*-a.

To je jedna neverovatna stvari Ta baza podataka imala je 3.5Gb fizičke memorije. Na žalost nismo mogli naći module od 2Gb ili su samo od pola Gb po kartici. Kranuli smo sa tim i pokušali da pravimo uporedni test. Otprilike *baby-unik* može da da maksimalno pola gigabajta za keš za bazu podataka. A prava mašina, odnosno Alfa može da de neograničeno. Odnosno, ograničeno je samo koliko čovek može da spakuje memorije u mašinu. Onda smo napravili dva konfiguracije *baby-unix* od pola gigabajta i pravu sa 3.5Gb memorije za bazu podataka. Merili smo kako se ponaša sistem kada sa cela baza od 3.5Gb spakuje u memoriju. Rezultati su bili više nego fascinantni za sve nas. Jer sve je to toliko jednostavno – pristup memoriji je još uvek jedno sto hiljada puta brži nego pristup disku. Svako zna da počeli pet ili deset milisekundi za disk sa 50 nano-sekundi za memoriju i dobija se ta vrednost od 100 hiljada. Je samo 50 000 puta brže. Za jednostavne transakcije sa TPCA, TPCB- dobijali smo na jednom procesoru koji uopšte nije bio u tom trenutku najbrži neka brojke kao što je tristo transakcija po sekundi sa uniprocesorom. Bez ikakve optimizacije, samo onake da vidimo koliko dobijamo.

Onda smo napravili test pri kome je u memoriji bilo 20% baze. Dobijali smo pedeset ili tako nešto, transakcije u sekundi. Rezlog je bio prost – mašine mora da ide na disk. Kao prvo mora da čeka podatke a kao drugo mora da troši snagu na transfer iz memorije na disk. To je bilo prosto tu se na vidi prava razlika. Pesignut je faktor 3 ili 5. Zatim smo napravili kemplikovane šplikacije koje u stvari čine realan život – jer realan život nisu proste aplikacije tipa TPCA ili TPCe – i tu recimo upit baze podataka na toj *baby* konfiguraciji od samo pola gigabajta memorije traje četiri sata. A kada je cela baza podataka u memoriji – traje šest minuta. Smeete ame dolli de fascinantnog broja faktor 40 ili 50. Megao sam da očekujem brojke tipa 200 ili 400 ili 1000 jer još uvek govorimo da je pristup memoriji u odnosu na pristup disku brži jedno sto hiljada puta. Mi smo dobili samo 50 puta. Cela poenta je što je za taj tip aplikacija *mainframe* nemoćan. On završene na 2Gb i posle toga mora da ide na disk. Mi dalje samo tražime veće memorijske kartice i dobijamo ubrzanje. Upit koji je trajao dva dana, ovde traje pola sata. Upit koji traje na *mainframe*-u mesec dana, ovde traje jedan dan – 8 pola dana.

U pravodu stvari koje su jednostavno bile nemoguće – postaju moguće. O tome čovek nije mogao ni da sanja. A to je sve bio uniprocesor. Na velikoj mašini kada čovek gleda tirugi ili treći ili peti procesor, performanse neće biti značajno bolje. Jer ti isti procesori samo vrte pačleve u krug i čekaju na podatke. U slučaju Alfa procesori jednostavno zvižde kroz memoriju.

Računari: Kali mi neki primer povećanja brzine sa Alfom u odnosu na klasične rešenja sa sistemima.

Pol Krneta: Teško sam sa kojom brzinom može ova sadašnja Alfa – koja samo što nije dočekala inovaziju nova generacija – da prebriše memoriju ide kroz memoriju i može da prebriše dvesta megabajta po sekundi. To je jedna tipijalne operacije – uplavanja nečega u memoriju. One upiše osam bajtova odjednom. To je šokantna stvar kada vidiš da te tri i po gigabajta ona prebriše sa nekih petnaest dvadeset sekundi. Sken baza podataka veličina tri i po gigabajta sa optimiziranim algoritmom za keš traje oko dvadeset sekundi. Ne mašini koja radi sa diskovima to traje satima. Čovek čuje te brojke i teško može da zamisli šta to znači. Primenjeno na kola faktor pedesat može da znači pedesat puta dvesta ne set, što iznesi oko deset hiljada kilometara ne set. To su već neuporedive stvari. Kada dodu nova generacije procesora, kažu – brži je 50%.

Računari: Kada neko hoće na može da zamisli da leti.

Pol Krneta: Kada neke stalne razmišlja u kategorijama hodanja, to su onda hodači koji ne sto metara hodaju trideset sekundi. Oni koji trče postižu 30 dolari trčati postižu 15 a svetski rekordan davet i po sekundi. Oni koji iste uopšte na razmišljaju kako je to moguće da sa sto metara prede za vreme koje traće od sekunde. To je jednostavno nova dimenzija.

Računari: Odnos Alfe i mainframe-a nije protivnički?

Pol Krneta: Poenta uopšte nije protiv *mainframe*-a. Jednostavno – došle je nova tehnološka generacija. Bolje je od onoga što *mainframe* može da ponudi. *Mainframe* ima jedinu stvnu osobinu – *input* i *output* mu je bio jako dobra strana. To je činjenica. Mašina otvori 50 kanala. Svaki je po 5 Mb/sec. Te je stvarno vrlina vrednost. I to je stvarno možna stvar. Tu čak neme šta puno de se razgovara. Unix je tek sada dorastao da može da zahteva da kranu tim putem. Sa tolikom memorijom to sada više nikoga ne interesuje *input/output*, to je prevaziđeno za neke aplikacije. Sve je u memoriji i diskovi nem se vrte u prazno u stvari. I to je ono što je lepo. Jednostavno ama zaobišli ceo taj problem. Čovek kupi najobičnije diskove koji se kupuju u samoposluzi i – reći.

Računari: Kada smo prošli put razgovarali, govorio si Alfu sa Pentiumom. U međuvremenu samah dobija PowerPC. Da li ti se menjaje sveju Alfu sa PowerPC?

Pol Krneta: (smaje se) **Nel Problem je u tome što je PowerPC i dalja 32-bitna mašina. Kao drugo što su joj performanse znatno slabije. Govorimo negde o 70-80 SPECmarks. Cela familija PowerPC će biti vrlo rasceprkana. Imaće 32-bitnu mašinu, imaće 48-bitnu imaće 64-bitnu kroz neko vreme. Čovek hoće da kupi *entry-level* PowerPC – to je 32-bitna mašina, spora, sa kojo može da kaže. Aha, brži je od Apple-a. Ako hoće da razvija pravi 64-bitni softver koji u sebi nema neke limite mora da kupi drugu mašinu. To je kao kada vozi u prvoj brzini a za drugu brzinu mora u garaži da promeni menjač. Hoće u trci opet mora da promeni ceo menjač u garaži. Sa Altom je šepa stvar što čovek kada kupi najmanju Alfu koja postoji na ovome svetu ona je smesta celokupno 64-bitna sa svim što tome pripada. Svaka, recimo Alfa za koju je uradim *tuning* ubedena je da ima nilljade gigabajta memorije na raspolaganju. Jednostavno takva je konfiguracija da čovek može da piše bilo kakav softver. U ovom trenutku limit je, zbog veličine page-a od 8Kb 4 terabajta virtuelnog prostora po procesu. Znači ako ima sto procesa svaki vidi 4 terabajta. Alfe u ovom trenutku podržava *page-size* do 64 kb i se 64 kb odiči će na 56 adresnih linija na 56 linija virtuelnog adresnog prostora bez ikakvih promena u hardveru i to je se tri *page table entry* nivoa. Ako se pojavi neko ko može da stvarno pokaže i dokaže da mu treba više dodaće se i četvrti *nivo page table entry* i može se preći i više od 64 bita. Može da se ode na 80 bita virtuelnog adresnog prostora po procesu.**

Računari: Da li softver pisan za ovu generaciju ostaje kompatibilan i sa sledeće generacije?

Pol Krneta. Poenta je u tome što softver koji već danas radi na Alfi, koji ima te strašne ograničenje da može da vidi samo četiri hiljade gigabajta virtuelnog adresnog prostora po procesu da to nije limit nego je samo trenutna implementacija koja će vremenom da se promeni naviše. Jer do sada niko nije čak ni prišao blizu tom limitu. A softver koji je pisan za ovu generaciju ostaje apsolutno kompatibilan se novim generacijama. Operativni sistem OSF sada proverava da li je program pokušao da upiše nešto iznad 43 bita. Ako je probao kaže ne, to je greška. Nova generacija će da kaže program radi. Jer kompajler ne zna se ograničenja. Kompajler je potpuno 64-bitni. Danas napisana aplikacija radiće i kroz dvadeset godina na Alfi koja, recimo ima sto hiljade terabajta RAM-a ili neku potpuno suludu brojku i moći će potpuno da se koristi. Softver koji je danas pisan radiće kroz deset godina. To je lepota.

Na velikoj mašini ti isti procesori samo vrte palčeve u krug i čekaju na podatke. U slučaju Alfe – procesori jednostavno zvižde kroz memoriju.

Baza podataka, ona će da raste. Tek sada moramo da očekujemo eksploziju baza podataka kada čovek može da ima takvu performansu. Jednostavno – Alfa + baza podataka = *high-end* mašina. Jer, limit na postoji. Današnja Alfa će da kaže izvini ja ne mogu da idem preko 43 bita. Kroz dva godine će da kaže Super, čovek je došao do pedesetog bita adresnog prostora i radi i dalje. Stvar je sigurna. Ako je sa njeku Alfu aplikacija pravilna, ona će da kaže ne mogu – ali softver funkcioniše. Nema razlike u kompajlerima i u razvoju softvera. To je ono što se kaže, da veličina ne menja filozofiju. Da li je ikada iko mogao da kupi mainframe za 10 000 maraka i da ne njemu razvija softver? Ne. Niti je mogao niti će moći. A ovde svako može da kupi Alfa PC se OSF/1 i da razvija softver za najveću budućnu Alfu i da uzme tu disketu i da je kroz deset godina gume u mašinu i kaže – to radi. U tome je sadržano ono što se naziva *investment protection*. A, recimo se PowerPC će biti haos 32-bit ni 64-bitni, biće stalne migracije. Kao se SUN-om na primer. Sanos 4.1 Solaris 1. Solaris 2. Stvari koje u principu nisu nikta fundamentalno novo donele a samo uzimaju novac za migracije. Sa OSF/1 puno narednih godina ćete biti mirni.

Računari: Proces savladivanja otpora i predrasuda – vrlo pametnih ljudi profesionalaca u oblasti – traje.

Pol Krneta: Malo njih je uopšte do sada razmišljalo o 32 bita kao o limitu. U stvari malo njih je do sada taj limit dosegio. Postoje neki koji su van njega, toga su vrlo dobro svesni i već su odavno na Alfi. Ali za većinu ljudi je problem da nešto što je do sada funkcionisalo shvate kao budući limit. Pojavila se Alfa i odjednapput su sve 32-bitne mašine koje su važile kao *state-of-the-art* postaje, hm nešto što još uvek funkcioniše ali znamo da mu je mandat istekao. Kao predsednik u odlasku. Još neko vreme i onda mora da napusti svoju kuću bilo koja boje. I prva reakcije velikog broja pametnih ljudi je. To mi ne treba. I većini stvarno ne treba. Ali ekskoma kaže. Nikada nema viška memorije, nikada nema viška MIPS-ova i megaherca! Bilo koja aplikacija može da pojede sve MIPS-ove ovoga sveta sve gigabajte ovoga sveta. Mi

Sve je te toliko jednostavne – pristup memoriji je još uvek jedno sto hiljade puta brži nego pristup disku.

kada bukvalno imamo kupca koji su se rešili za Alfu kao za strašnu stvar. *Unix* je vrlo fleksibilan i ako se nekome ne sviđa DEC-ova firma, može da ode do Packarda ili bilo koga drugog i u roku od par dana ili nedelja radi se dalje. Jer *Unix* je vrlo kompatibilan od platforme do platforme. Stvar se u principu rekompelira i ide se dalje – u tome je draž *unix*-a. Firma koja je zamenila svoje malo veće mašine sa Alfama dabila je u roku od poslednjih 12 meseci, sa pet ili šest sati, kvantni skok za faktor 60 ili 100. I tu se niko ne oseća nelagodno po sistemu. Jeo bože šta ćemo sa ovim suvišnim MIPS-ovima ili megaflopimal? Gde ćemo to da iskoristimo!?

Računari: Svi se odmah odomaće, familijarizuju.

Pol Krneta. Naravno! To se iskoristi momentalno. Jer čovek nešto što je radio osam sati sada radi pola sata, petnaest minuta, pet minuta. Sada već hoće da bude interaktivno.

Računari: Opšti trend su interaktivne aplikacije

Pal Krneta: Sve aplikacije gde se očekuja neka interaktivnost od mašine – tu se drastino gade na snagu procesora. Recimo simulacija analognih kola traju par sati ili rutinarje štampane ploče traje par sati. Neki put i preko vikenda. Sada, ja sam inženjer koji hoće da proba tu varijantu na deset načina. Deset puta po vikend je deset vikenda pola godine ja već prolila. A ja hoću da kažem – klik klik – i da se pojavi ova ili ona varijanta sa statistikom ostalo je deset nepovezanih čvorova ostalo je šest nepovezanih čvorova ili osmo ja nijedan. Hoću da radi se mnom u realnom vremenu. Uzmi *update* Lotusove tabele. Neću da se on muči se tim pošle klika tri minuta, hoću da on to uradi odmah. Znači. interaktivnost! Pravilo je prosto. Sve sto traje duže od par sekundi nije interaktivno. Kada nešto postane toliko interaktivno da bude ispod sekunde onda čovek doda samo novi *layer* softvera i odjednom to postaje pet puta duže.

I onda se opet traži novi procesor. Kada pogledamo šta je urađeno sa procesorima u poslednjih deset

Konkurencija je ignorisala Alfu, jer to je najbolji marketinški potez – ne dozvoliti svojim kupcime de ukapiraju da postoji Alfa.

godina, vidimo da su se ubrzali bar se faktor hiljadu, ako ne i više. Sada je Alfa brža 150 puta od VAX a i nikako ne dobije kazne za prebrzu vožnju. I doći će nova se 1200 MIPS-a koja će biti 500 puta brža od VAX-a. Pa šta onda. To će kroz deset godina biti stara mašina.

Cene memorija po megabajtu stalno padaju. Cena po kartici ostaju manje više konstantne. A na kartici se stalno nađe kako vreme protiče dva ili četiri puta više memorije. Ono što je pre osam ili deset godina bile monstruozna mašina – MS-DOS mašine od 640 kb – pa kome to danas treba? Eila su to vremena kada je *high-end* mašina imala megabajt memorije. Danas svi kada govore o PC-ju govore o baram 4Mb 6Mb to je normalna stvar 16 Mb za malo jače mašine. A to je već faktor dvadeset ili trideset. I niko se ne unbuduje.

Računari '90 smo skidali i šitali kako treba da izgleda konfiguracija personalnog kompjutera. 1Mb memorije, 100Mb disk, ber 17 inčni monitor rezolucije 1200 sa 1000 umrežanje i 32-bitni procesor

Pol Krneta: Sve 32-bitne mašine osjeću sudbenu na 2 gigabajta memorije. Danas je standard 16 megabitni čip znači 2 megabajta memorija. Ceo MS-DOS staja u 1/4 čipa danas. Kroz nekoliko godina doći ćeme u situaciju da će ceo sdisesni prostor jedne *baby-unix* mašine stati na par čipova. Pa će stati na jedan čip, e zatim na pole čipa. Samo zbog jedna prosta stvari što ta mašina ne može da adresira više od 32 bita.

Računari: Znači pojam baby-unix vesun je sa 32-bitni adresni prostor?

Pol Krneta. Da. To još uvek na znači da *baby-unix* ne može da se koristi u realnom životu ali bilo koja investicija dugoročna ili čak srednjeročna postaje besmislena. To obavlje odradeni posao: ali to je istorijski *pipeline* gde ono što je bio današnji *mainframe* postaje miniračunar sutra postaje PC prekosutra postaje ko zna koji nikto neće od nakosutra. To je taj šok. Alfa je poslala niz ličmu puno vibracija onima koji su odjednom shvatili gde su ostali.

Računari: Ako pretpostavimo da u svetu danas ima desetak firmi se snažnim razvojnim timovima od hiljadu do dva ljudi, koji bi mogli da se upuste u razvoj nove generacije i stvarnuje vodeće pozicije u domenu masovnog računarstva, u čemu je razlika koja je baš DEC dovela do tog rešenja e nekog drugog?

Pol Krneta: Ceo problem se danas svodi na nekoliko u principu jednostavnih stvari. Prvo je da razvoj samog procesora kao procesora danas mora da traje vrlo kratko. Drugo, svaki procesor koji se danas pojavi mora da bude vrlo dugovečan da bi se isplatio – to je nešto u šta vredi investirati. Treće procesor sam po sebi je na smam de kažem da je trivijalna stvar. All to je lakši deo zadatka – sama arhitektura. Četvrto – tehnologija. Zamislimo recimo Alfu koja je divna koja je 64-bitna i tako dalje all je spora jer DEC nema poludprovodničku tehnologiju. Zamislimo tu situaciju koja je paralelogalne. Svi bi rekli – baš je stvarno divna i čak se na bi ni borili protiv nja toliko. Rekli bi – divna je stvar all šteta što nje brza. All sada, pošto je koncept dobar posto se vidi da je brza i vidi se da se u tom pravcu tek kreulo svi sad traže mene. Dosta ih je tako rekli. Postoje mane koja su akademske ili ograničenja koja su stvar implementacije. Ono što je bitno, ta velike firme mogu sve što ja potrebno da o-

Razvoj takve stvari zahteva viziju pre svega. I to je ono što razlikuje dečake od muškaraca. Mora se imati vizija da bi se razvilo nešto što traje puno godina.

kupa all razvoj takve stvari zahteva viziju pre svega. I to recimo razlikuje dečake od muškaraca. Mora se imati vizija da bi se razvilo nešto što traje puno godina. Jer mi znamo firme koje su napravile nove računare koji su ubrzo zatim zavalili ne otpadu. Ne zato što su loši, već zato što je nedostajalo vizije. 36-bitni računar na primer.

Procesor je tek prva karika u celom lancu. Ako je on loš niko više neće da se obazre na njega. On mora sam po sebi da buda odličan koncepcija mora da bude odlična. Implementacija u smislu dizajna mora da bude odlična. Tehnologije mora da bude odlična u prevodu mora da bude performantan. Jer performanse ja danse „name of ine game"

Zalim, softver je bitna stvar – procesor sam po sebi na može da odluči. More da ima softver za sebe jer operativni sistem sve više spade u *commodity* OS mora de bude odličan, da omogućava sve i da nikada ne bude prepraka. More da bude ona komponenta koja omogućava de to funkcioniše a ne da buda limit – je sam važnij od bilo čega. Ja sam samo sluga koji omogućava da nešto bito nako zamisli rad beeprekorno. Aplikacije danas rešavaju stvar i to je svima jasno. Zbog toga je DEC uložio puno napora da migracije aplikacije buda više nego jednostavna. Recimo mi u Minhenu u centru imamo mem od dva dana po projektu migracije. Kod nas dode softverska kuća sa bilo

keg Unix-a se Ultrix-a sa HP-a, Sun-a IBM-a. U velikom broju slučajeva mi posao završimo za dva dana. Postoje izuzeci naravno all cela poenta je u tome što je OSF toliko standardan OSF je spojio dve stvari. Govorim samo o OSF/1 na Alfi. Unix je u principu dosadna stvar. Osnovna varijenta je odnalne stvar. Ono što ga čini lepim su aplikacije i funkcije koje ima OSF – realno vreme 64-bitno adresiranje i cela priča koje se tim ide. Sama migracije je trivijalna stvar. I nakon toga kada čovek sebere sve te komponente, tek onda može da se govori o uspehu nečega.

Ono što je DEC-u pomoglo – a to je stičaj okolnosti – je to što je VAX kao arhitektura bio na kraju svoga životnog veka. DEC je znao da mora da izađe sa novom generacijom mašina na tržište. To je jednostavno

To je istorijski *pipeline*, gde ono što je bilo današnji *mainframe* postaje minikračunar sutra, postaje PC prekosutra, postaje krš koji niko neće od nakosutra.

bilo pitanje opstanka. Da nije Alfe, DEC bi propao. Jer znao bi staru tehnologiju. All kod DEC-a je bilno da je uvek vizije postojala. Recimo PDP je živeo svojih dvanaest petnaest – dvadeset godina. VAX je uveden 1978. godine i živi i dan-danas. Još ima jako puno kupaca koji kupuju iskljućivo VAX jer VAX kao tehnologija, kao procesor radi na 110 MHz. To je ravnovalna stvar CISC mašina koja radi na 110 MHz.

Računari: All i drugi veliki proizvođači u oblasti računarstva su na zalasku 32-bitnog sveta. Zašto onda DEC uspeva s njihovi razvoji na uspevaju?

Pol Krneta: Neke mora da budoi prvi DEC je prvi uveo mini računare. Dec je prvi uveo 32-bitne mašine. Neka su firme 1984. godine posle uvođenje VAX-a, još uvek prodavale svoje 16-bitne mašine kao *high-end* mašine. I, uostalom neka ima viziju a neko je nema. I to je stvar kulture. DEC je razvijao svoje RISC mašine od 1984. godine. Napravio je jednu pa je odbacio, jer je VAX bio suviše sposoban – „nećemo da kvarimo". Pa sa pojavile druge pa treća RISC mašine. Ovo je četvrta generacija internio razvijanih mašina. To je otprilike horor scenario kaoel u zvaničnoj trci na javnoj sceni trči VAX a iza scene se zagreva onaj koji je jači od njih svih ali – nećemo da kvarimo scenu. A uvek je postojala „shadow" mašine koje je mogla od proguta VAX, u bilo kom trenutku, po performansi po koncepciji All jednostavno vreme je bilo se VAX. A sada je vreme za Alfu i ona se pojavila!

IBM se pojavio se Unix-om 1990 godine. IBM pre toga uopšte nije imao Unix. IBM je tada napravio svoj 32-bitnu RISC RS/6000 mašinu. Pre toga imao je neke poluzaja koji su strateški biti slepa ulica. Definicija strateške stvari je – nešto što smo danas napravili se da funkcioniše u narednih 10, 15 ili 20 godina. A nešto što se brljavi mrljavi avake dve-tri godine je do kraj kratkovidosti. Ako imaš viziju, vidiš sto kilometara.

DEC je znao de mora de izađe sa novom generacijom mašine na tržište. To je jednostavne bilo pitanja opstanka. Da nije Alfe, DEC bi propao

Kada govorimo o novoj generaciji 64-bitnih računara i govorimo samo o performansi to je uvrede za oko tu viziju. To jeste performansa, all je i puno više i u domenu programiranje, elegancija kojom se dolazi do novih rešenja. Postaju moguće stvari koje su do sada bile nemoguće.

Računari: Da li postoje igre za Alfu?

Pol Krneta: Sve igre koje postoje na Unix-u mogu se vrlo brzo konvertovati. To je posao od bukvalno nekoliko minuta. Ko ima *source*, samo ga prekompilira na Alfi i to je to.

Ono o čemu se sada naveliko priča je da će se pojaviti precodiiac sa Intela za Alfu. A tome bi se radovali svi koji takvo pitanje mogu da postave. Jer na Intelu postoji puno igara na MS-DOS- u Windowsima.

Računari: Znači, to još uvek nije zvanično objavljeno?

Pol Krneta: Ima u laboratoriji, ima se velike kupe, all još to nije postao zvanični proizvod. Jer to je sada postala jedna životna realnost i potreba se pojavom Alpha Windows NT.

Računari: All zar Alfa NT nema u sebi taj prevodilac?

Pol Krneta: Ima all mi nećemo svaki put kada uzmamo neku aplikaciju da je prevodimo. Mi hoćemo da Alfa radi u turbo režimu a na se tegom oko vrata. Želimo oil imamo taj prevodilac koji će oil prevede jednom i pri tom izvrši određena optimizacija. Pitanje igara svodi se na pitanje – kako de moju aplikaciju se Intela smesta imam i na Alfi. MS-DOS i Windows svet je pun firmi koje propadaju, a je imam upravo aplikaciju koje mi je potrebna od firme koja je propala. Šta od radim? Prevodim je na Alfu.

Računari: Šta se najnovija dogada u svetu Alfi?

Pol Krneta: Govorima sada o nekim novim Alfa ma. Na jesenjem COMDEX-u se pojavile jedna nova Alfa. Kutijica veličine Alfa PC 320 MHz. 4Mb keš. Ta je mašina jedno 80% brža od najbrže alfe koje se sada prodaje. Stoji na stolu. To neće biti naposredan proizvod već je direktna najava. u malom enoge šta stiže. Kada su Alfa u pitanju novi procesori će dolaziti kao ludi. Novi ima na sebi *quad issue* – četiri pipelinea 2+2 dva nivoa keša. Time je rešena jedna vrlo jednostavne stvar. Imati dve nivoa keša na čipu je direktne posledica razvoja onoga čipa od 500 i od 1000M Hz od pre godinu-dve dana. Tada su to probali i rekli – ako nema dva nivoa keša na čipu zaboravimo ga. Čip je sam po sebi brz, all dok stigne da memorije nastupa kraj priča.

Nikade neme viška memorije, nikade neme viška MIPS-ova i megaherca! Bilo koje aplikacije može da pojede sve MIPS-eve ovoga sveta, eva gigabajte ovoge sveta.

Samim tim je sada DIGITAL razdvojio dve stvari. Može da razvije sumanuto brze procesore sa svojih 300 400 500 1000 MHz i dalje. Oni sami za sebe su mislene dovojni jer imaju već dva nivoa keša. Manja je potreba i za onim *off-chip* kešom koji je danas 4 ili 8 Mb – jednostavno keš koji je na čipu je sinhron se Alfom na 300 MHz. Sasoi su se dočaćima ruke razvezale. Dobili su slobodne ruke da naprave što je moguće brži procesor sam po sebi. O okruženju ćemo mi oil se brinemo a ova grupa koja razvija procesore je slobodna da ide dalje. To je još uvek C-MOS C-MOS na 320 MHz.

U DEC-u spremaju OSF multiprocesorske mašina. Jer VMS je već odavno multiprocesor. OSF je još uvek *single* procesor. negde mi marta kreće se oko 32 *multiprocesorom*

Računari: To znači da su sve velike Alfe O-penVMS?

Pol Krneta: Uprave to. Sada čekamo novi kvantni skok. Do sada jedino što je se *unix*-om na unipracesor mašini bilo moguće to je 350 transakcija po sekundi. Očekujemo faktor 3 do 5 se *unix*-om na multiprocesorskoj mašini. Plus novi procesori sa 1200 MIPS-a po procesoru.

Računari: Kako konkurencija reaguje na Alfu?

Pol Krneta. Dešava se oil svako de svoju ponudu za neki sistem. Tu onda padna neki benčmark, uzmu se štoperice u ruke i vidi se koliko ko može da unati. Naravno, Alfa je obično bila brža. Pojavljivala sa kao relativno nova mašina stara godinu dana, pogotovu sa Unix-om. Posle prvih uspeha Alfa obično su govorili: Dobro to je sporadično. mi ćemo sledeći put pobediti. Sve igre su to počelo ponavljati konkurencija je ignorisala Alfu. Jer to je najbolji marketinški potez – na dozvoliti svojim kupcima od ukapiranju de postoji Alfa – po sistemu – *Kept in dark, fed on shit*. Onoi je na mnogim mestima ljudima objašnjeno šta je Alfa i tu dolazi do brutalnih rezultata po konkurenciju. Tipična reakcije konkurencije je. Možemo li sada de damo poslednju ponudu? I daju popust od nekih sedamdeset

Nastavlja se na strani 60

ADACOM
ENGINEERING
Tel: 629-233, 337-367, 344-492 Fax: 629-233
Kneza Miloša 9

	KORISTIMO	CASE ALATI ID/AD/AG CASE	GENERATORI RELACIONIH BAZA PODATAKA MAGIC II POST 4 GL
NUDIMO			
KONSALTING			
PROJEKTOVANJE			
IZVOĐENJE			
UVOĐENJE U RAD			
REVERZIBILNI INŽENJERING			
ODRŽAVANJE			
PLATFORME			
UNIX			
VMS			
WINDOWS, NT			
NETWARE			
OS/2			
DOS			
CTOS, BTOS			
ŠKOLOVANJE			
ISSA			
CASE ALATI			
MAGIC			
WINDOWS			
NOVELL			
UNIX			
KOMUNIKACIJE			

brzo, efikosno, kvalitetno, povoljno

VELIKO FINALE

Najveća nagradna igra u istoriji časopiaa „Računari", sa nagradnim fondom od 45.000 DEM, koja je trajala od maja do kraja oktobra ove godina, završena ja 14. novambra velikim finalam pred kamarama TV Baograd, u amlaiji „Zabavna nedalja".

U predvidenom roku na adresu naše redakcije je stiglo oko pet hiljada koverata se kuponima. Kao što emo obećali, izvlačenje dobitnika je bilo javno – obavljano je 14 novembre 1993, u okviru emisije „Zabavna nedelja" prvog programe RTS Direktan televizijski prenos izvlačenja osam premija je počeo u 16 35, dok je ostalih 99 dobitnika izvučeno nakon što su se kamere ugasile, u studiju 9 Radio-televizije Srbije u Košutnjaku

IZVLAČENJA MOŽA OA POČNE

Da bi sve proteklo u skladu sa propozicijama, brinuo se žiri u sastavu 1 **Dejan Ristanović**, programer i publicista (predsednik), 2. **Zoran Životić**, direktor razvoja u „Jugodati", 3. **dr Milan Kovačević**, direktor u IMTEL-u, 4. **Veselin Jevrosimović**, visenik firme COMTRAD, 5. **Veena Jeremić**, komercijalni urednik u časopisu „Računari"

Od preko dvadeset naših sponzora, završnici nagradane igre su prisustvovali **Lenka Lazarević** (direktor firme LS DATA), **Žika Vodeničer** (direktor marketinga u firmi COMPU TER BEOGRAD) **Danko Jevtović** (direktor u firmi JUGODATA) **Donat Greber** (direktor firma PC PLUS). **Boris Damjenović** (direktor razvoja u firmi OSA), **Veselin Jevrosimović** (vlasnik firme COMTRAD), **dr Milan Kovačević** (direktor firme IMTEL)

Nakon uvodnih reči o nagradnoj igri koje je dala **Veena Jeremić**, komercijalni urednik „Računara", vođenje izvlačenja je preuzala iskusna voditeljska ekipa „Zabavna Nedalja" – **Nina Mudrinić, Dragana Katić** i **Jovan Pavličak. Za** prvih osam dobitnika srečne ruke je bile mlade glumice **Ivana Mihić**, dok su ostalih 99 koverata izvukli prisutni korisnici Sazama – **ana, veca i darona**

Žiri jaste imao pune ruka posla, ali je sve proteklo glatko – od 107 izvučenih koverata svi su bili regularni – u mnogima je uz 4 uredna kupona bio i džoker!

Već kroz tri dana, 17 novembra, dobitnici premija su se okupili u redakciji „Računara", da preuzmu svoje nagrade Daim Beograđana, po izuzetno lošem vremenu su stigli i dobitnici iz Pirota, Srbobrana, Novoga Sade Jedino su poručnika Janka Stijepovića iz Podgorice, služba, udaljenost i vremenske neprilike sprečili da dođe(naravno, stigao je za nekoliko dana). Gotovo svi su poveli i pomoćnike – jer trebalo je iznet toliki hardver! Kao što vidite, okupljanje sedmorce najsrečnijih je zabeležila kamera studija Tešić, a mi smo iskoristili priliku da saznamo nešto više o svakom od njih.

OSMORICA NAJSREČNIJIH

Najviše sreća je imao **Jovan Kostić** iz Pirota, dvadesetšestogodišnji inženjer elektrotehnike. Njemu je pripala specijalna novčana nagrada firma IMTEL, garantno pismo u vrednosti od 5 000 DM za kupovinu opreme po sopstvenom izboru i do sada se Jovan bavio računarima, a sa mašinom iz svojih snova koju će moći da dobije u IMTEL-u, pokušaće da

Izvlečenje može da počne: Ivana Mihić, Jovan Pavličak, Veena Jeremić, Dragana Katić, Nina Mudrinić

Sreća je bila u rukama mlade glumice Ivane Mihić

početne ozbiljnji posao na razvoju i prodaji softvara.

Super-premiju firme ComTrad, računar 486DX na 50 MHz i štampač HP DESKJET 500, je dobio još jedan inženjer elektrotehnike, magistar **Vladimir Marković** iz Beograda. Donedavno je g Marković radio na istraživačkim poslovima u Institutu „Mihajlo Pupin", a trenutno je zaposlen u Direkciji za razvoj PTT saobraćaja.

Premija koju je poklonio Info D – računar 486DX na 50 MHz – je odradovale jednog stanovnika Studentskog grada – **Srebrenka Čopa**, dvadesetšestogodišnjag apsolventa Šumarskog fakulteta, inače iz Teslića. Na svom starom računaru 286 intenzivno je radio u programima *Symphony* i *WordPerfect* a sada će, sa novom 486-ticom, imati priliku da se okuša i u *AutoCad*-u

Nagrade firme PC Plus – računara 386DX / 40 MHz je iznenadila **Boruta Stanojevića** iz Beograda. Pre no što je došao po svoj novi računar, nazvao nas je i s nevericom priso da li je baš on taj. Ipak, nadeo da će „od toliih nagrada dobiti barem kutiju disketa" I g. Stanojević se profesionalno bavi računarima – radi na njihovom održavanju u preduzeću „Informatika" u Beogradu.

Računar koji je poklonila Jugodata – 386DX na 40 MHz, stigao je zasta u prave ruke, **Zoranu Radakoviću** iz Beograda. Zoran je mladi istraživač, inženjer mašinstva i saradnik Mašinskog fakulteta u Beogradu, gde završava postdiplomske studije iz oblasti mašinskih materijala. Posebno se radovšao RAM-u od 4MB na svom novom računaru, jer mu je to nedostajalo za komfornji rad

Laserski štampač HP LaserJet 4L, poklon firme Computer Beograd, otišao je u Podgorici **Dobitnik, poručnik Janko Stiljepović**, je sevršio Vojno-tehničku akademiju i radi u Centru se automatsku obradu podataka Komande 2A. Programire u Clipper-u, a interesuju ga i mreže i animacije Sa ovim čudom od štampača moći će da se ozbiljnije bavi i odradom i pripremom teksta za štampu

Video VGA karticu koju je poklonila OSA je dobio **Dejan Petrović** iz Novog Sada, inače redovan čitalac Računara, ali prvi put učesnik nagradne igre. Apsolvent je Mašinskog fakulteta i bavi se programiranjem u C-u

Osmu premiju, džepni računar Psion firme LS Data, dobio je **Zdravko Šandin** iz Srbobrana, vlasnik firma za automatsku obradu podataka. Računare čita od prvog broja, a želje mu je da uskoro postane i čian Sazama.

ČESTITKA I UTEHA

Ubrzo su i ostale nagrade, u obliku hardvera i softvera, napustile redakciju Računara i našle put do svojih srečnih vlasnika. Neke kao što su kursevi i prelpiate, tek treba da donesu radost čitaocima kojema su pripale Sve u svemu, stotinu sedam naših čitalaca se ovoga puta radovalo, ali biće još prilika i za one koji su ostali razočarani! To je nagovestio glavni i odgovorni urednik „Računara", **Jovan Regasek**, završavajući veliko TV finale nagradne igre ovim rečima. „Dobitnicima nagrade čestitam, a svim učesnicima želim više sreće sledeći put. Sledeća šansa, sledeća nagradna igra, sledeći maj i nisu tako daleko Želeo bih da se zahvalim svima koji su uzeli učešće u našoj nagradnoj igri, a naročito sponzorima, jer bez njihove dobrote na bi bilo ni ove radosti Nagradne igre dolu i produ, a časopis „Računari" ostaje, da vas informiše o zbivanjima iz sveta računara i obučava za njihovu primenu "

Srećni žiri: Dejan Ristanović, Zoran Životić, dr Milan Kovačević, Veselin Jevrosimović

Predstavnici sponzora: Lenka Lazarević (LS Data), Žika Vodeničar (Computer), Danko Jevtović (Jugodata), Donat Greber (PC PLUS), Boris Damjanović (OSA)

Najsrećniji na okupu (s leva nadesno): Zdravko Šandin, Dejan Petrović, Srebrenko Čop, Jovan Kesić, Vladimir Marković, Zoran Radaković, Borut Stanojević.

KO JE DOBIO NAGRADE

Premije

1. **INTEL** – Specijalna računarska nagrada, garantno pismo u vrednosti od 5.000 DM za kupovinu opreme po sponzorskom ključu
JOVAN KOSTIĆ PIROT

2. **CentriTest** – Računar 486DX / 50 MHz, local bus, 8 MB RAM, local bus video sa 1 MB RAM-a, hard disk 210 MB, FDD (5,25"/3,5"), SVGA color (Daewoo), vertikalno tower kućište, tastatura, miš PLUS štampač HP-DESKJET 500
VLADIMIR MARKOVIĆ BEOGRAD

3. **Info ID** – Računar 486DX / 50 MHz, ISA, 4 MB RAM, Trident 1 MB, hard disk 210 MB FDD (5,25" 3,5") SVGA color (Hyundai) vertikalno tower kućište, tastatura, miš
SREBRENKO ČOP BEOGRAD

4. **PC Plus** – Računar 386DX / 40 MHz, 4 MB RAM, VGA 512 KB, hard disk 130 MB (Seagate ST3144A), mono VGA monitor (HC) FDD (5,25" 3,5") tastatura, desk kartica Soundblaster „Zebra" sa zvučnikima
BORUT STANOJEVIĆ, BEOGRAD

5. **Jugodata** – Računar 386DX /40 MHz, Trident (8900) 4 MB RAM, hard disk 85 MB, mono VGA (SCAN), mini tower kućište tastatura, miš
ZORAN RADAKOVIĆ, BEOGRAD

6. **Computer** – Laserski štampač HP LaserJet 4L
JANKO STIJEPOVIĆ, PODGORICA

7. **SRA** – video kartica (Trigevision 16P PAL) – PAL grafička mašina za konverziju VGA u video do rezolucije 800x600 u 256 boja i VGA prikaz do 1024x768 u 256 boja
DEJAN PETKOVIĆ, NOVI SAD

8. **LS Data** – Mrežni terminator PISON tehnol 3, 256 KB RAM
ZDRAVKO ŠANDIN, SREBORNA

Štampači

9. **Adamson** – Štampač PANASONIC KX-P1123i (24-igličn)
ALEKSANDAR JOCIĆ, BEOGRAD

10. **SoftProjekt** – Štampač PANASONIC KX-P1123i (24-igličn)
ALEKSANDAR MEDIĆ, BEOGRAD

11. **„Računar"** – Štampač CITIZEN 124 D (24-igličn)
ZORAN SIMONOVIĆ, BEOGRAD

12. **SAGA** – Štampač Star LC 20 (9-igličn)
ALEKSANDAR ŠUKARSKI, NOVI SAD

13. **MB Sistem** – Štampač HYNDAI HDP-600 (A3 24-igličn)
ALEKSANDAR MIJANKOVIĆ, VALJEVO

14. **IBC** – Štampač HYNDAI HDP-600 (A3 9-igličn)
IVAN BELOVIĆ, BEOGRAD

15. **ASYS COM** – Štampač EPSON LX-100 (9-igličn)
BRANISLAV NOVKO BEOGRAD

Grafičke kartice

16. **MICRO AIR** – Grafička kartica DIAMOND Stealth 24 (16,7 miliona boja) S3 grafički čip rez 1280x1024 sa 1 MB RAM i prateći softver
ALEKSANDAR ŠUPLJAC, BANJALUKA

17. **MICRO AIR** – Grafička kartica DIAMOND SpeedSTAR, Hi Intel (32 000 boja) TSENG grafički čip rez 1024x768 sa 1 MB RAM i prateći softver
VLATKO MANTEV BEOGRAD

UPS

19. **Microdata** – On-line UPS uređaj od 650 VA
NEMANJA BAMARIŠIĆ BEOGRAD

18. **KR DATA** – UPS – 750 napredno napajanje sa dodatnim redundan 700 W
KOLARIĆ MILAN BEOGRAD

Monitori

20. **MP Biro** – Color VGA monitor (BTK, B')
ANITA TODOROVIĆ, VRNJAČKA BANJA

Telefoni

21 28. **MIC** – Personalni telefon
ZORAN STOJSAVLJEVIĆ, SUBOTICA
ZORAN JANKUJEVSKI, KIČEVO
IGOROA MANSIMČEV NOVI SAD

24. **Telmotejam** – Personalni telefon
BORA JANCVLJEVIĆ, BEOGRAD

Pisači sto

25. **Ibro-Coop** – Pisači sto
MIHAJLO DODIĆ, NIŠ

Softverski paketi

26. **Sprint** – vBASE IV
BORIS PETROVIĆ, BEOGRAD

27. **Sprint** – Turbo Pascal for Windows
STANISLAV POPPAŽIĆ, KRUŠEVAC

28. **Sprint** – Turbo Pascal 6.0 za DOS
ŽIVAN BEZANOVIĆ, ZEMUN

29. **Sprint** – Quattro Pro
STEVAN IVETIĆ SUBOTICA

30-31. **OSA** – Autodesk CA LAB
IVAN MILANOVIĆ, BEOGRAD
DEJAN PANTOVIĆ RECKUNAS

32-33 **OSA** – Autodesk CHAOS
MIODRAG STANKOVIĆ, ZAJEČAR
KATJA RANGELOV DIMITROVGRAD

34. **MiSoft** – MS DOS 5.0 Upgrade
PETAR ŠOKARDA, PANČEVO

35. **Softland** – Softland Quattro Pro 1.0
JELISIJA PANTELIĆ-BABIĆ BEOGRAD

Kursevi

36-46. **Tehnoteam** – Kurs po izboru
DUŠKO BATOĆANIN, NOVI BEOGRAD
STEVAN BUNIĆ BEOGRAD
KAMPE LEOKO, NOVI BEOGRAD
TRAJAN PETROVIĆ, ROGATIĆ
MLADEN MRKIĆ BANJALUKA
BORAN SIMIĆ BEOGRAD
ZORAN MARINKOVIĆ, BEOGRAD
VLADAN ORLUIĆ, BEOGRAD
ZORAN DANKOVIĆ, BEOGRAD
DANIJEL SUELICA, NOVI SAD

46-51 **Mikro Anima** – AutoCAD ili CorelDRAW
ALEKSANDAR ĐAKIĆ NIŠ
VALENTINA KOSTIĆ, NIŠ
GORAN RAČIĆ, BEOGRAD
TOMISLAV LINKAŠEVIĆ, ZEMUN
IVAN KADOR, BEOGRAD
RADICA MILINKOVIĆ, ZAJEČAR

52-56 **Institut „Boris Kidrič"** – kurs po izboru
MILORAD SIMIĆ BEOGRAD
DOBRICA NIKOLIĆ, SUPSKA
PRISMAS BLAGIĆ, PRIŠTINA
LASLO SRKOVICA, SUBOTICA
ZORAN MILINČIĆ, SREMČICA

Miševi

57-61 **Sprint** – MB
ALEKSANDRA STOJNOVIĆ, KRUŠEVAC
MIROLJUB DRAGIČEVIĆ, NOVI BEOGRAD
MOMIR MARTINOV ZRENJANIN
VLADAN SRKOVIĆ, LESKOVAC
DARKO MILOŠEVIĆ, BEOGRAD

62-66. **Tehnoteam** – MB
MILJKA ADAMOV STARČEVO
STOJAN SIMIĆ, NOVI BEOGRAD

64. **MP Biro** – MB
MIGLINSKI PETER, SUBOTICA

Diskete

68-70. **SAGA** – Diskete
KOJANOVIĆ DARKO, ČAČAK
BEJAN PEŠIĆ, ZAJEČAR
BRANISLAVA PRAMCUSKI, KRINDA
NIKOLA PAVLOVIĆ, NOVI BEOGRAD
ZORAN VITEZ, BEOGRAD
LENDITA QEHAJA, PRIŠTINA

71 72. **Tehnoteam** – Diskete
SNEŽA ĆUBLOŠEVIĆ, BEOGRAD
ALEKSANDRA MIJANKOVIĆ BEOGRAD

Filteri za monitor

73-77. **Sprint** – Filter za monitor
JOVAN POLJAČIĆ, NOVI BAR
ZORAN VUKOVIĆ, ČAČAK
KRISTIJAN VOJNIĆ, SUBOTICA
MIODRAG LATINOVIĆ, NOVA PAZOVA
MIODRAG POPOVIĆ BEOGRAD

78. **Adamson** – Filter za monitor
BRANKO ŠAMANDIJA, BAČKI GRAČAC

Knjige

78-80 **Institut „Boris Kidrič"**
IVAN KRALJ, BEOGRAD
ALEKSANDAR LUKIĆ, SREMSKI KARLOVCI
GRADIMIR MINIĆEVIĆ, NOVI BEOGRAD
BRISIJAN ĐUROVIĆ ŠVRLJIG
DEJAN JEDINĆ, NOVI BEOGRAD
VLADIMIR MILOŠEVIĆ, BEOGRAD
RADOMIR KOSTIĆ NIŠ
ZVEZDAN KARAĆ, VRNJA
NEBOJŠA ĐORĐEVIĆ, BEOGRAD
DMITRIJE BROĆIĆ, NOVI SAD

80. **CET**
BRANKO ŠAMINJANOVIĆ, BATOČINA

Ostale nagrade

80. **CET** – Drijač za papir i ribon
JOVAN VIDIĆ NOVI BEOGRAD

81. **CET** – Predavanje za WP 5.1 (Windows
DUŠAN PANIĆ-VASIĆ JUKOVIĆ, ZEMUN

Prepiate na Sezam

85. Besplatna prepiata na Sezam
JASENKA STEFANOVIĆ, BEOGRAD

86-94 Prekogračnje prepiate na Sezam
DRAGICA ZORIĆ, BEOGRAD
NIKOLA GRDZANIĆ, BEOGRAD

95-97. Transatlantska prepiata na Sezam
RADOSLAV JANJIĆ, VALJEVO
NADA KORLITIĆ, RUMA
MILOMIR MILOVANOVIĆ, ZRENJANIN

98-101 Jadranskomorska prepiata na Sezam
MILAN KOZLOVIĆ, NOVI BEOGRAD
VLADIMIR PETROVIĆ BEOGRAD
GORAN DETOJIĆ, NOVA PAZOVA
BORAN DRAŽIĆ RUMA

Prepiate na „Računare"

106-110. Besplatna prepiata na „Računare"
ZVONIMIR RODIĆ, REBNIK
ALEKSANDRA KOSTIĆ BEOGRAD

104-105. Pokaganitska prepiata na „Računare"
PREDRAG STOJKOVIĆ, NOVI BEOGRAD
PAVLE TMUŠIĆ, ZEMUN

106-107 Transatlantska prepiata na „Računare"
BORAN MLADENOVIĆ, NOVI BEOGRAD
MATIJA BRKOVIĆ, BEOGRAD

PRVE LASTE

Istina je: Pentium PC će stvarno da promeni PC svet. Možda ne odmah, ali tokom sledeće dve godine sigurno. Promena će ići mnogo dalje od ubrzanja rakaikulacija u programima za tabelarne proračune ili osvežavanja slika na ekranu monitora. One sa neće izražavati samo prostim zbirom impresivnih tehničkih vrednosti. Pentium tehnologija će promeniti strukturu same kompjuterske industrije, jer Intel namerava da pobedi u trci sa RISC procesorima.

PC *Magazin* je testirao sedam prvih mašina bazirarih na Pentium procesoru kad i čudo koje dolazi iz kompanije NCR – računar koji može da istovremeno koristi 16 Pentium procesora, i onda razvija brzinu od 1440 MIPS. Čekajući da prve ploče sa Pentium procesorima stignu i u naše krajeve da vidimo šta su iz novog procesora izvukla najveća imena u PC svetu – IBM Compaq DEC Unisys Hewlett Packard Acer i ALR. Ovoga puta nećemo ulaziti u pojedinačne opise mašina. Zadržaćemo se samo na rezultatima testova i generalnim utiscima.

DVE KLASE MAŠINA

Testirane su dve klase mašina, radna stanica i server. S obzirom da je najzanimljivije bilo videti kolika je stvarna snaga Pentium procesora, napravljen je filter po kome su mašine razvrstane. U klasu radnih stanica su grupisane mašine koje su imale minimum 12Mb RAM-a, hard disk od najmanje 300Mb 15-inčni monitor sa adekvatnom SVGA grafičkom karticom i Intelligent DOS 5.0 i Windows 3.1. Za servere je zahtevano da imaju najmanje 16Mb RAM-a, hard disk kapaciteta između 500Mb i 2Gb i *NetWare 3.11*.

Test za radne stanice je ispitivao ponašanje aplikativnog softvera. U tu svrhu su neke garniture testova dodate u već postojeće testove – *Hardware Benchmark Test* i *Winbench 3.11*. Svi testovi se izvršavaju automatski i precizno kontrolišu određene poslove u programima *Excel 4.0*, *Superbase 2.0*, *AmiPro 3.0* i *CorelDraw 3.0*. Po prvi put su izvršenja merena se i bez keširanja diska. Bio je korišćen *SmartDrive* iz programa *Windows 3.1* i za njega je odvojeno 2Mb RAM-a. Odabran je upravo ovaj keš program zbog njegovih visokih performansi ali i zato što je verovatno najrasprostranjeniji među konsnicima koji prilagođavaju softverskom keširanju hard-diskova.

Serveri su testirani u mreži pod *NetWare 3.11* sa 32 mašine klase 386/20MHz. Cela mreža je pokrenuta putem *Netbench 4.0* testa, koji koristi datoteke promenljive dužine i promenljive veličine blokova, da bi se simulirala mrežna aktivnost.

Očekuje se da PC računari bazirani na Pentium procesoru zapljusnu tržište u tri talase tokom sledeća tri meseca. Ovih prvih sedam mašina su samo prethodnica. Sve su uglavnom projektovane na osnovu postojećih koncepata za 486 sisteme. Ali, i pored ovoga, demonstrirale su zavidan Pentium potencijal u odnosu na ulaženo vreme i novac. Uskoro ćemo se sresti se većom količinom mašina od različitih proizvođača. One će, takođe najverovatnije biti bazirane na 486 projektima, a kolika će snaga moći da izvuku iz Pentium procesora zavisiće od sposobnosti inženjera tih kompanija i pažnje koja će biti posvećena projektovanju. Očekujemo da će druga generacija računara bazirarih na Pentium procesoru biti prikazana krajem 1993 godine. Biće to računari projektovani „iz temelja" da bi zadovoljili glad Pentium procesora za podacima.

Ovi sistemi će imati skupu i visoko optimizovanu keš memoriju, memoriju I/O sistem i ostale podsistema potrebne da bi se u potpunosti iskoristila snaga koju je Intel ugradio u Pentium procesor.

Pentium sistemi koje smo testirali su nedvosmisleno, brzi i snažni, u proseku su pokazali za 75% bolje performanse nego trenutno najbrže mašine zasnovane na 486DX2/66 procesoru. Međutim, dolaskom novih Pentium mašina iz Pentium procesora će se isceditl mnogo više – masovnu isporuku i početak prodaje treba očekivati tek u 1994. godini.

Za desktop mašina koje bi se uglavnom koristile za CAD ston inženjerstvo ili finansijske proračuna, povezane u mrežu radi *file-sharing*-a, ova hronologija uopšte nije bitna. Bilo koja mašina iz prve ili naredne druge generacije biće sasvim dobra i vrlo brzo će se isplatiti. Ali oni koji traže stvarno visoko-performantne servere, tj. mašine koje će isceditl i poslednji atom snage iz Pentium procesora moraće malo da sačekaju.

KAKO DOSTIĆI RISC

Za svaku novu generaciju procesora koja se pojavljivala, bilna su bile brzina i kompjuterska snaga. Reporterski izveštaji o Pentium čipu govorili su optimistički o povećanju kompjuterske snage u odnosu na 486DX2/66 za oko 4-5 puta. Ovi izveštaji nisu bili pogrešni, ali su u stvari bili pogrešno interpretirani. Čelni do pet puta veća brzina se odnosila na operacije u pokretnom zarezu što nije najvažnija performanse procesora. Brzina izvršavanje operacija u pokretnom zarezu je proizašla iz Intel-ove želje da brzinom *floating-point* matematike konkuriše RISC procesorima. Naime RISC procesori su postali vrlo velika prelnje Intelovom CPU tržištu, jer je se sada svaki RISC procesor bio brži od Intel-a za čitav red veličine u izvršavanju operacija u pokretnom zarezu. Pentium-ova brzina rada se operacijama u pokretnom zarezu je sada 4-5 puta brža nego kod najbržeg 486 procesora, ali još uvek je sporija od većine RISC procesora.

Prve Pentium laste: Hewlett Packard NetServer LM5/60 i Compaq DeskPro 5/66M

Na veliku želost PC korisnika, mali je broj poslovnih PC aplikacija koje koriste operacije u pokretnom zarezu. Neki programi, kao što su *AutoCAD*, sadrže rutine se rad u pokretnom zarezu ali ipak većina PC aplikacija živi u mnogo sporijem svetu celobrojne matematike. Međutim, to će biglada da se promeni. Pentiumova snaga se pokazala vrlo inspirativnom se inženjere softvera, tako da su mnoge softverske kuće počele da razmišljaju o poboljšanju svojih proizvoda, dodajući im sve više i više rutina za rad u pokretnom zarezu. Takođe, mnogi su iskoristi se rekompajliranjem svojih aplikacija što će biti nephodno da bi se Pentium upotpio u punoj meri.

Pentium na nov način definiše i tržište radnih stanica. Unapređenje rada se pokretnim zarezom će se pokazati kao vrlo važan strateglјski potez jer će na osnovu toga Intel pokušati da progura sisteme bazirane na Pentium procesoru na tržište tehničkih radnih stanica, na kojem za sada dominiraju sistemi bazirani na RISC procesorima. U tom kontekstu konkurentna cena Pentium procesora će bachi iz obilјima razmišljanja potencijalne kupce RISC radnih stanica.

Ključni element ovih promena biće, takođe, i dugo najavljivani Microsoft Windows NT operativni sistem koji bi trebalo da se pojavi krajem 1993 godine i koji će se izvršavati na širokoj paleti kompjuterske platforme: od tradicionalnih RISC sistema do PC-ja. Tako bi se dobila kompatibilnost zasnovana na operativnom sistemu a ne na hardveru.

SA DVE BRZINA

Pažljivji čitaoci će zapaziti da neki proizvođači koriste Pentium koji radi na 60MHz, dok ostali koriste brži 66 megaherci Pentium procesor. Ova disharmonija je, u stvari, odraz problema u samoj proizvodnji Pentium procesora, kao i želja Intela da proda što više završenih čipova. Nemoguće je obezbediti stoprocentnu proizvodnju savršenih čipova. Uvek će nekoliko čipova, koji se u toku proizvodnju nalaze na specijalnoj silicijumskoj pločici (*silicon wafer*) „pasti" na jednom ili više rigorozanih čip-testova. Nije ni čudo kada se pogleda se kolikom mikroskopskom preciznošću je Pentium raden. Procesor 486 je baziran na 1 mikrometarskoj tehnologiji, dok su „vodovi" koji spajaju elemente u Pentium-u široki jedva 0,8 mikrometara. Da bi se dobila realna predstava o ovim dimenzijama, treba reći da je 1 mikrometar milionti deo metra, ili oko 1/15 debljina dlake ljudske kose. Kombinovati minijaturizaciju na tom nivou sa proizvodnjom velike količine čipova je pomalo obeshrabrujući posao. A Intel je, želeći da bude i dalje predvodnik u izradi čipova i procesora, postavio sebi zadatak da svakih 18 meseci udvostručuje broj tranzistora na jednom čipu. Tako je Intel sebio 3,1 miliona tranzistora u Pentium čip, za razliku od 1,2 miliona tranzistora koji se nalaze u 486 procesoru.

Pentium procesori koji rade na 60MHz uopšte se ne razlikuju od onih koji rade na 66MHz. Jedina razlika je u tome što Pentium procesori koji rade na 60MHz nisu prošli sve testove kojima je Intelfen procesor koji treba da radi na 66MHz. Intel je i ponudio sve ne baš

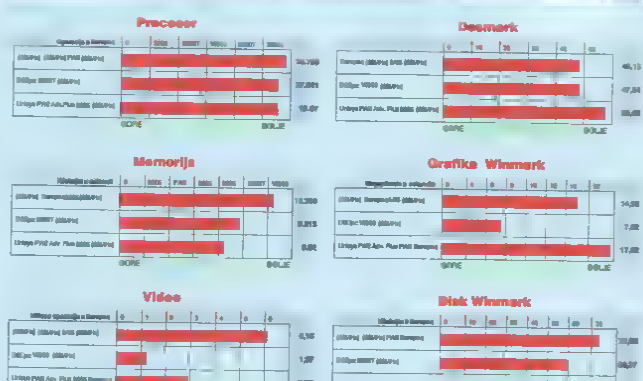
savršene procesore nastrpljivim proizvođačima. Glavni *motiv* je bio da se što pre počne se izradom mašina zasnovanih na novom procesoru. U praksi, nema nikakve razlike između ova dva procesora, a Pentium procesori koji rade na 60MHz su potpuno ispravni čipovi. Takođe, u toku testiranja smo naišli na vrlo zanimljive pojave. *Pri testiranju grube procesorske brzine* naravno pokazalo se da su Pentium procesori na 66MHz se oko 10% brži od procesoru koji rade na 60MHz. Međutim, se druge strane se pokazalo da je mašine koju je sastavio Unisys, PW^2 Advantage Plus, u procesu bile brža od Compaq i DEC mašina koje su radile na 60MHz. To samo pokazuje koliko je Pentium procesor mlad čip i koliko su inženjeri neiskusni da iz njega izvuku svu svagu. Za početak jedan savet: zanemarite frekvenciju na kojoj radi Pentium procesor u mašini koju kupujete.

Ograničenje koja nameće tehnologija 486 računara se kada se Pentium mašine koriste kao serveri ili kao samostalne radne stanica nisu bazirana na slabostima samog procesora. Znatno sporiji I/O u kombinaciji se malom keš memorijom i sporom RAM-memorijom čine da procesor izgleda spor. Mašine prve generacije pokazuju očiglednu želju proizvođača da optimizuju kritične podsistema – što je više pažnje posvećeno optimizaciji u toku projektivanje celog sistema, to smo mogli da vidimo i veća poboljšanja. Primer odličnog projektovanje lošim se izvlači maksimum iz Pentium procesore je Compaq-ova TriFlex arhitektura, sa brzom i širokom 64-bitnom CPU magistralom i sistemno širokom 128-bitnom memorijskom magistralom, te adekvatnom *bus* logikom. Kao ni projektovanje solidnog Pentium sistema, ni izrada matične ploče na kojoj će se nalaziti Pentium procesor nije jednostavan posao. Na mnogim pločama smo videli puno džampera i prototipova raznih specijalizovanih čipova, kao i neočekivano veliki broj raznih kratkospojnika (žica koje se naknadno postavljaju tokom projektovanja i izrade ploča, čime se ploču) i raznih ostalih dodataka se ploča koje nisu pažljivo projektovana. Proizvođači čiji su standardni AT kompjuteri bili uvek u tehnološkom vrhu i ovde su pokazali istu preciznost i kvalitet.

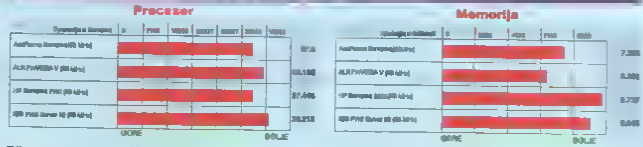
KAD SE ZABORAVE BROJKE...

Kada zaboravimo *na benchmark* brojke, vrlo je važno podrobnije pogledati sistem koji je projektovan da zadovolji potrebe kupaca. Na primer, klasa mašina koje su predstavljene kao idealni serveri je pokazala slabu stranu u mogućim proširenjima. Tako je, na primer, DEC u svom PC 560ST, za koji se planira da se prodaje i kao radna stanica i kao server ponudio samo pet mesta za dodavanje internih jedinica (flopi, hard drajv

WINDOWS RADNE STANICE



SERVERI



Dijagrami predstavljaju standardna testovi časopisa PC Magazine se testiranje hardvera: Hardvare Benchmark Tests, v7.01; Windows Benchmark Tests (Winbench), v 3.11 Za real-word performanse korišćeni su sledeći programi: Ami Pro v3.0, CorelDraw v3.0, Microsoft Excel v4.0, Superbase v2.0 PS.

PENTIUM PC

Pentium je najbrži, trenutno najmoćniji mikroprocesor u PC svetu. Superskalarne operacije *pipeline* i 64-bitne operacija nuda postojane superiorne performanse, bez obzira da li je kompjuterski sistem projektovan da služi kao *mrežni server* ili kao *Windows* radna stanica.

Toplota je i dalje briga kako projektanete tako i instalera Pentium mašina. Pojedini sistemi koji se pojavljuju imaju montirana rotorasta hladnjake neki imaju mini ventilatore, a neki, opet, koriste oba sistema za hlađenje.

60MHz i 66MHz su trenutno maksimalne brzine za sistema bazirane na Pentium procesoru, ali sam čip će uskoro da radi i na celih 100MHz.

Radna stanica III servera pitanja je sad? Postoje sve Pentium mašina su tako moćne da su već sada za oko 78% ispred najbolje 486 mašine.

IBM PS/2 server 95 je prvi PC koji je prešao barijeru od 30 miliona procesorskih instrukcija u sekundi.

Compaq Deskpro 6/S6M poseduje 128-bitnu sabirnicu, što mu daje prednost nad ostalim mašinama, bar što se tiče komunikacije sa memorijom. Deskpro je sposobna da komunicira se memorijom 20% brže od bilo koja postojeće Pentium mašine.

Preiaz se 486 na Pentium se može jednostavno izvesti kod mašine baziranih na matičnim pločama koje dozvoljavaju zamenu/dodavanja procesora, biće jednostavan i se pojavom P24T *overdrive* procesora, e rešenje je naravno, i zamena cele matične ploče Svaki napredak se 486 (ili slabijeg procesora) na Pentium ima svoje odbre i loše strane. Najzad najbolje rešenja je kupovina kompletna mašine bazirane na Pentium procesoru.

Pentium vam izgleda malo sporo? Nudimo vam dva mogućnosti da razmislite o multiprocesorskom radu koji će koristi 2 – 16 Pentium procesore II da sačekate do 1995 godine kada se očekuje pojava P6 procesora

PCI (*Peripheral Component Interconnect*) arhitektura je novi sistem lokalne sebirnice. Odlično sarađuje se Pentiumom i nudi potencijalnu prednost nad VESA lokalnom sabirnicom.

Najbolji Pentium sistemi kombinuju široku sabirnicu se efikasnim eksternim kešom i sposobnošću se vrlo brze *read/write* zahteve.

strimer) i svega šest EISA slotova se proširenja (video, kontroleri itd.) kao i vrlo slabašan 256W sistem napajenje. „Digital" prodaje ovu mašinu kao server se male mreže ali se mi na bismo prijatno osećali se tako limitiranom mašinom. Neke druge mašine za razliku od ove, imaju puno mesta za dodavanje novih uređaja, dovoljno slotova se umetanje novih kartica i sasvim dovoljnu električnu podršku se njihovu buduću ulogu. Na primer, IBM-ovo i Hewlet Packard-ove kućišta je sasvim prikladno se ulogu servera

Video performanse su bila sledeća stavka na koju se gledalo prilikom testiranje funkcionalnosti. Kod većina mašina predviđenih za servera nije se vodilo puno računa o video performansama, jer se njihovi monitori obično koriste za praćenje stanja na mreži. Naravno, video mogućnosti servera i nisu tako bitne, ali je kod radnih stanica video sposobnost jeden od najznačajnih elemenata u celom sistemu. Pošto se se Pentium radna stanice predviđa da se koriste najčešće kao grafičke radne stanice i za CAD odnosno slično izdavaštvo *Windows* video performanse su bile pod posebnim prismotrom. Od tri radna stanica koja su testirana, Unisys-ova mašina je koristila čuvanu ATI Mach 32 grafičku karticu i ona se logično našla na prvom mestu se 17 7 miliona *winmark*-ova. Ti nije najbrža grafička kartica koju smo dosad videli, ali je rezultat ipak dostojan poštovanja. Uz Unisys, vrlo blizu je bio i Compaq se

NADGRADNJA: JEDNOSTAVNIJE NEGO ŠTO MISLITE

Ko ne bi poželeo da oseti port prstima snagu mašina zasnovanih na Pentium procesoru? Međutim ako posedujete nedavno kupljen znači skoro nov 486 sistem sigurno se nalazite pred dilemom da li da u njega upregnete najmlađi i najbolji PC mikroprocesor današnjice.

Tržište već nudi nekoliko alternativa sa različitim nivoima performansi i jednostavnosti instalacije. Da se ne biste opremili pogrešnom opremom ili, još gora PC-jem koji će zastariti pre vremena, morate vrlo pažljivo da odvagnete svaku alternativu.

Najbolja strategija prelaska na Pentium je, naravno, kupovina sistema formiranog oko Pentium-a. Pošto procesor poseduje 64-bitnu magistralu sa podatka samo će sistem i koji su projektovani za Pentium CPU iskoristiti u potpunosti njegovu snagu. Pentium procesor ima dvostruki paljalan za interdžer sku aritmetiku, dva keša i radiosinjranu jedinicu za matematičke operacije u pokretnom zarezu (FPU) – sve to zahteva i veći i brži eksternr keš nego 486 procesor.

Nadgradnja postojećeg PC-ja je jedna od mogućnosti koja se može ostvariti uz pomoć dva metoda. Možete da zamenite matičnu ploču koja sadrži procesor potrebna logička kola, I/O i, često, video i [illegible] ga alternativa je da instalirate Pentium CPU modul koji sadrži procesor a često i keš memoriju, naravno, ako je proizvođač vaše matične ploče predvideo takvu mogućnost.

Poseb kupovine Pentium PC-ja sledeći korak u praćenju tehnologije bio bi zamena matične ploče za [illegible] će da prihvati Pentium OverDrive procesor. Postoje dva razloga zašto se to ne preporučuje: prvi je to što se, na žalost, 238-pinski procesor baziran na Pentium tehnologiji sa kolnim imenom P24T, neće pojaviti pre kraja 1994 godine. Drugi razlog je što postoje indicije da ta strategija nadgradnje neće

(opet) matičnu ploču sa Pentium procesorom, ali koje bi imala najvažniji čip-set a sigurno i *VL-bus* ili PCI konektore koje vaša ploča trenutno nema. Dobili biste i poslednju verziju BIOS-a dizajniranog baš za taj Pentium sistem, i matičnu ploču na kojoj je nelen problem grejanja samog procesora.

Mnogi PC proizvođači koji prave module sa nadogradnju matičnih ploče uključujući tu ALR AST, Compaq Dell i IBM već su kreirali module bazirane na Pentium procesoru. Prednost takvih modula je u tome što su napravljeni sa najvećom pažnjom i sposobnošću da iz Pentium-a izvuku maksimum. Ali jedno upozorenje: iako su mnogi već ponudili takve module samo modul baziran na Compaq TriFlex arhitekturi radi sa 64-bitnom sabirnicom. Ostali moduli su bazirani na 32-bitnoj sabirnici, što znači sporiji rad sa memorijom.

Najbolji CPU moduli neće imati samo Pentium procesor nego i novi kristalni oscilator (zato što Pentium procesor radi na 60 i 66MHz) eksternu keš memoriju i sistemsku memoriju. U većini slučajeva biće potrebno da zamenite i BIOS bilo kao softversku nadogradnju, ako se on nalazi u *flash*-memoriji, ili zamenom ROM-čipa.

Poslednji i sada već vrlo sumnjiv, način je ku- [illegible] će da prihvati Pentium OverDrive procesor. Postoje dva razloga zašto se to ne preporučuje: prvi je to što se, na žalost, 238-pinski procesor baziran na Pentium tehnologiji sa kolnim imenom P24T, neće pojaviti pre kraja 1994 godine. Drugi razlog je što postoje indicije da ta strategija nadgradnje neće

pružiti punu Pentium snagu jer P24T procesor neće podržavati tekozvani *burst* mod čitanja/pisanja. Niko na kaže da će P24T procesor biti slabiji ili da ga ne vredi čekati. Problem je u tome da se procesor još uvek nije pojavio i da se na može sagledati da li će postojeće ploče koje se već uveliko prave i prodaju se odgovarajućim podnožjem stvarno i podržavati očekivani procesor. Drugim rečima, da li će biti sposobne da iskorste sve njegove mogućnosti. Neki proizvođači neće moći da pruže podršku P24T procesoru sa potrebnom količinom keš memorije i ostalim neophodnim čipovima. Osim toga, mnoga ploče sa 238-pinskim podnožjem su zasnovane na 486SX i 486DX procesorima a ne na robusnijem 486DX2 procesoru koji ima Intel-ovu specifikaciju.

Još uvek se ne zna da li će P24T imati dvostruku *integer* jedinicu kao i pravi Pentium procesor. Zna se jedino da P24T neće imati 64-bitnu magistralu jer će se koristiti kao nadogradnja u 16-bitnim i 32-bitnim sistemima. U međuvremenu Intel planira da proizvede i ostale Pentium-bazirane procesore koji će verovatno imati dvostruki (DX2) i trostruki (DX3 33/99MHz) radni takt.

Neki proizvođači su, međutim, prauseli Intel-ovu 238-pinsku strategiju napretka. Na primer, Dell ube- [illegible] sa mašine sa oznakom 4000XE Server Series nudi pravi Pentium CPU modul za one koji ne žele da čekaju. ACER koristi specijalnu verziju 238-pinskog podnožja koje prihvata ne samo P24T nego i standardni 486 procesor.

grafičkom karticom QVision, koja je sakupila nešto manje od 15 miliona *winmark*-ova. Krajem leta Compaq je najavio poboljšanje svoje QVision kartice, tako da može da radi sa rezolucijama do 1280x1024. DEC PC 560ST sa grafičkom karticom (Cardinal) baziranom na S3 čipu postigao je svega 8 miliona *winmark*-ova.

Novina koja će značajno da poboljša upotrebljivost matičnih ploča se Pentium procesorom je tekozvani *flash-BIOS* sistem. Sa prvim Pentium mašinama

pobo[illegible]šanja BIOS-a su i potrebna i poželjna. Flash-BIOS omogućuje korisnicima da naprave nadgradnju (*update*) postojećeg BIOS-a, i to jednostavnim pokretanjem odgovarajućeg programa sa diskete dobijene od proizvođača. Znači nova verzija BIOS-a se u stvari dobija od proizvođača matične ploče i kao što se vrši *update* softvera, tako se vrši i *update* BIOS-a. Pet od sedam testiranih mašina su koristile *flash-BIOS*.

Vrlo bitan element kod ove nove generacije procesora je i celokupno hlađenje. Pentium čip za vrlo kratko vreme dostiže temperaturu od preko 100 stepeni (C). Što je još gore Pentium-bazirani sistemi sadrže razne elektronskim elementima, koji takođe znače znatne količine toplote. Da na spominjemo hard diskove velikog kapaciteta, koji su takođe značajan činilac zagrevanja. Pregrejani čipovi ne uništavaju samo sebe, već i okolne komponente na matičnoj ploči. Ni matična ploča ne

NOVI STANDARD: PCI

PCI (*Peripheral Component Interconnect*) arhitektura o kojoj se ove godine najviše govorilo izgleda da postaje stvarnost. Iako je Intel pokrenuo ovu tehnologiju i imao prvi čip-set na tržištu, ni ostali nisu sedeli skrštenih ruku već su krenuli sa sopstvenom implementacijom.

Pošto je Intel prikazao svoj 82430 PCI čip-set, ostali proizvođači su počeli sa proizvodnjom sistema koji u potpunosti iskorišćavaju sve mogućnosti Pentium-a a bez značajnih izmena postojeće PC arhitekture. PCI arhitektura je projektovana da radi u saglasnosti sa postojećom ISA ili EISA sabirnicom.

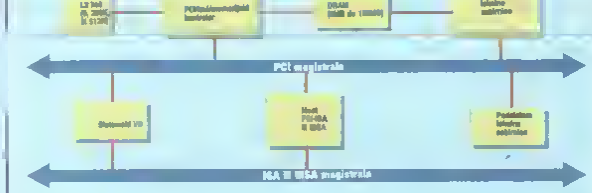
PCI povezuje grafički video čip, LAN I/O, SCSI i osnovne I/O komponente preko odvojene sabirnice. U današnjim PC računarima ove komponente su bazirane na ISA, EISA ili MC (*micro chanel architecture*) sabirnicama. Sa PCI arhitekturom ove komponente mogu da rade na 32-bitnoj sabirnici na 33 MHz. ISA sabirnica je široka 16 bitova i radi na celih 8MHz. EISA i MC standardi, iako 32-bitni, rade na samo 8MHz odnosno 10MHz, respektivno.

82430 PCI set se sastoji od Intel-ovih 82434LX PCMC (*PCI/Cache/Memory Controller*) i 82433LX LBA (*local bus accelerator*) i od PCI/ISA ili PCI/EISA

„mosta" preko kojeg se povezuje na glavnu sabirnicu. Prednosti PCI arhitekture su očigledne kada se uporede sa VESA VL-bus specifikacijom.

Iako su VL-bus (lokalna sabirnica koja se danas najčešće sreće kod sevremenih 486 matičnih ploča) i PCI gotovo identični na nivou propusnosti podataka (130Mb i 132Mb u sekundi respektivno) VL-bus arhitekturi dodatni kontroler sabirnice za ne-video uređaje i limitiran je na maksimalno tri *local-bus* periferna uređaja koja rade na 33MHz. Za brzine veće od 33MHz, moraju da se dodaju stanja čekanje da bi se komponente uspešno [illegible] periferija. 82430 PCI set podržava 512Kb sistemsku keša sa (*burst*) statičkim RAM-om.

PCI omogućava proizvođačima da integrišu ne samo video nego i ostale uređaje, tako što su LAN i SCSI uređaji direktno na lokalnu sabirnicu. Dok *VL-bus* danas omogućava da se (najčešće) ubrza PC grafika, PCI arhitektura ima potencijal da jednog dana postane standard za sisteme u kojima će raditi u poredno više periferih uređaja, i to maksimalnom brzinom mikroprocesora.



INTELOVI PLANOVI ZA SLEDEĆU GENERACIJU PROCESORA

Uz cenu gušvu koje je pratila pojavu prvih Pentium mašina, vrlo je verovatno da ćemo u na tako dalekoj budućnosti, zevati od dosade kada nam neko bude spominjao performanse Pentium-a. Znači, kao kada nam danas neko spominje procesore 8086 ili 286. U stvari naslednici procesora se ističnavim imenom P5 pod kojim sa Pentium razvijao su P6 i P7. Intel je već formirao timove inženjera za svaki čip, i oba tima rade punom parom na njihovom razvoju.

Intel je nagovestio samo par detalja u vezi sa sledećim generacijama x86 familije. Mada, sa istorijom kao vodilijom pronicljivi ljudi koji prate šta se dešava u Intelovim laboratorijama, mogu da sastave slagalicu i prikažu nam potencijalne mogućnosti budućih procesora.

Prvo, vitalni parametri procesora P6 biće stvarno impresivni. Broj tranzistora u P6, koji bi trebalo da se pojavi tokom 1994 godini popaće sa na 4-5 miliona. Poredeći taj broj sa sadašnjim ogromnim brojem od 3 1 miliona tranzistora spakovanih u Pentium, možemo da prognoziramo brzinu od celih 200 MIPS (miliona instrukcija izvršenih u sekundi) sa PC sistema a sve to na cenovni poređenje sa današnjih 112 MIPS koje postižu Pentium sistemi na 66MHz. Ako i P7 bude pratio dosadašnji 'vozni red' to će biti poslednji iz-

Da bi se osiguralo uspeh u P6 i P7 će biti ugrađeno mnogo više naprednih tehnika.

- Izvršavanje instrukcija bez strikinog reda. To će vrlo značajno da poveća performanse jer će natera ti čip da radi non-stop a ne kao do sada da čeka na rezultate odradenih operacija da bi započeo sa narednim
- Promena imena registera je put da se transparentno poveća broj registera. (Eksperti su se u globa lu složili da je cenovni hendikep x86 arhitekture mala količina registara.)
- Proširenje Pentiumovih superskalarnih mogućnosti preko granice izvršenja od samo dve instrukcije istovremeno
- Implementacija sistema paralelnog procesiranja nezavisnih procesnih jedinica, što će stvoriti mogućnost virtuelizacije više 386 procesora se jednim čipom

Ono što verovatno nećete videti u P6 ili P7 procesorima, bar ne u prvim primercima je integracija periferija kao što su video ili I/O elektronika, što je suprotno ranijem očekivanju. U kasnim 1990-im eksperti za mikroprocesore su predvidali da će vrhunski procesor oko 2000 godine u sebi sadržati sve potrebne čipove sa PC osim memorije. Taj nivo integracije bi obuhvatao sve počevši od logičkih i grafičkih čipova, od bežične komunikacije i multimedije

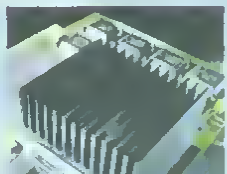
Danas te predviđanja na izgledaju verovatno Intel i ostali proizvođači procesora se usredsređuju na pakovanje što veće računarske snage u svoja vodeće procesore. Integracija je u drugom planu ali će se ipak koristiti da objedini starije procesore radi kompatibilnosti, i podržače čipove sa mobilne aplikacije

Poštujući ovu pravilo predviđanje čipove sa integrisanim periferijama u P6 ćemo videti onda kada budemo već špekulisali o tome na šta će ličiti P7 – verovatno u idućem veku.

Prestolonaslednik na kompjuterskom tronu

	486 familija	Pentium	P6 familija*
Broj tranzistora	1 2 miliona	3 1 miliona	4-5 miliona
Brzina klokа	25-66MHz	60-66MHz	66-132MHz
Performanse (na 66MHz)	54 MIPS	112 MIPS	200 MIPS
Veličina keša	8K	16K	32-64K
Datum pojavljivanja	1989	1993	1995
Veličina elemenata	1 mikrometar	0,8 mikrometara	0,6 mikrometara

*Očekivana konfiguracija bazirana na dosadašnjim saznanjima



Za skidanje visoke temperature, koja dostiže i 100 stepeni Celzijusa, proizvođači koriste ogromne ventilatore i masivne hladnjake

čekate sledeću generaciju Intel-ovih procesora. To će biti P6 i P7 koji sa projektu uporedo. Grubo predviđanje je da će mašina sa P6 procesorom da se pojavi krajem 1994 godine.

Za one koji su odlučili da žive na granici računarske moći sledeća opcija je multiprocesorski sistem. Tu je za sada predstavnik AcerFrame 3000MP koji prihvata do 4 Pentium procesora, ili neki drugi kompjuter, kao AST Manhattan – čudovište sa 16 Pentium procesora. Ako vam i to izgleda slabašno e raspolažete neograniče-

nom količinom novca i želite maksimalnu snagu koju PC može danas da pruži (samo je pitanje da li je to još uvek PC) možda će vas zadovoljiti NCR 3555, koji ima na samo 16 Pentium procesora nego i 2Gb RAM-a i 56 meuta za moguća proširenja (diskovi, strimeri). Je, naravno, i šestocifrena cena. Samo obezbedite i odgovarajući operativni sistem koji će biti sposoban da obavlja simetrično multiprocesiranje da iskoristi svu memoriju i ostali (eventualni) hardver. *Windows NT* ima ambicije da bude jedan od takvih operativnih sistema, ali se sada može da opeluši samo 2 procesora istovremeno.

I na kraju ako niste baš venčani sa Intel-ovom x86 arhitekturom procesora. Ima mašina koje su bazirane na novim, procesorima kao što su Digital-ova Alpha serija i Silicon Graphics R4000 mašina. Ako se obistine Microsoft-ova obećanja da će implementirati *Windows NT* na te platforme možda ćete moći prvi put u životu da pokrenete svoje aplikacije i na nekoj ne-intelovoj mašini. Pri tome bi većina postojećih PC korisnika želela samo da se oseća malo komfornije i potpuno je zadovoljna sa poznatim x86 svetom.

Karakteristike Pentium sistema su, se sada daleko ispred svega postojećeg u PC svetu kako po performansama tako i po ceni. Mnoge kompanije su ponudile svoje sisteme po ceni koja je malo ispod $10 000, a prednjače IBM PS/2 Server 85 sa svojom sirupom Micro Channel arhitekturom i višeprocesorski AcerFrame 3000MP, bar što se tiče cene. Ako vam ova cena ne odgovaraju da li možda vredi da razmišljao o nadgradnji 486DX2 sistema? Mnogi proizvođači će uskoro da ponude module za nadgradnju 486 sistema. Međutim zbog brzine memorije i uskog grla I/O na 486 mašini nećete moći da izvučete maksimum performansi iz Pentium procesora, mada čete moći da povećate performanse za 30-50%. Uskoro će se verovatno pojaviti i nadgradnje matičnih (486) ploča, koje će moždа dovesti kompjutersku snagu mašina koja smo ovoga puta predstavili

Mogući prilaz je i zamenili postojeći procesor sa Pentium OverDrive procesor. Intel je prilično zauzet ove godine da bi razmišljao o 'malim kupcima. Ali ako imate 238-pinsko podnožje na matičnoj ploči onda zaćekajte da se pojavi T24P čip. On će moći da se utiče na u vaš postojeći PC i pruži vam edekvatnu snagu. Koliko edekvatnu? To je za sada nemoguće reći ali evolucija Pentium mašina će nam u tome pomoći. Neki eksperti veruju da će Intel početi da pravi Pentium OverDrive procesora za godinu ili više dana i da će na večini Pentium procesora podići radni takt na 100MHz ili više. Mada će takve nadgradnje 486 mašina i dalje patiti od nedostatka memorija i keš memorije i imati I/O probleme, procesorska snaga koja će se izvući iz Pentium procesora će imati vrlo dobar odnos cena/mogućnosti. Ako želite više snage od danas najboljeg PC računara zasnovanog na 486DX2/66 procesora, rešenje je jedna od ovih sedam mašina. Mi bismo bez oklevanja zakoračili u budući 100+ MIPS svet Pentiuma i zamenili svoje 386 i 486 mašine za neku od prikazanih a verujemo i vi. Jer Pentium stvarno dolazi – već je počeo da menja PC svet.

NA KRAJNJOJ GRANICI

Ako vam ni jedan od ovih sistema još uvek na izgleda dovoljno moćan, imate više mogućnosti da sačekate nekoliko meseci i vidite da li će se druge ili treća generacija Pentium mašina materijalizovati po za crtanom raporedu. Prema predpostavci jednog projektanta sistema mašine projektovane „iz temelja" i baziranе na Pentium procesora biće sposobne da isporuče dvostruko veću snagu od one koju pruža prva generacija Pentium sistema. Druga alternativa je da se

MI GREŠIMO...

SNIŽENJE

SM S&S TIM TALENATA

... što koristimo ovakav naslov da bismo prodali tako kvalitetan štampač kao što je HEWLETT PACKARD.

S druge strane, MI NE GREŠIMO što računamo na vašu obaveštenost o poziciji HEWLETT PACKARD-a na tržištu štampača.

I izvinite za naslov. Jer grešiti je ljudski, a štampati je HEWLETT PACKARD-ovski.

Vaš ComTrad

ComTrad Combinacija Tražena dugo

Genex apartmani, Vladimira Popovića 6, 011 - 222 41 51, 011 222 26 52, 011 222 41 39

"SEDEX", Beograd 011 - 766 180, 764 451 • SPRIST, Novi Sad 21 - 628 717 • MARIR, Rumu 022 - 421 288 • INFOTRADE, Priština 038 - 28 038 • COMTRAD, Kragujevac 034 - 510 929

Mi prodajemo računare . . .

Biosfera Computers

MAKEDONSKA 22/6 (Dom omladine)
Tel/Fax: (011) 3229-109; 3224-378; 3248-208 od 9 do 20

. . . uz moguće doplate za . . .

HDD 125Mb=160DEM; 170=210DEM, 210=290DEM, 250=340DEM, 340=490DEM

VESA LOCAL BUS IDE kontroler + SVGA Cirrus Logic 5428 1/2Mb=200DEM

KUĆIŠTA: midi tower=100DEM; tower=150DM; big tower=200DEM

COLOR MONITOR: SVGA 14"=330DEM, SVGA 14" PHILIPS =500DEM, SVGA 17" =1400DEM

. . . štampače, dodatnu opremu..

EPSON LX400 A4 9 pina (180 cps) YU, kabl=420DEM
EPSON LX100 A4 9 pina (240 cps) YU, kabl, traktor =500DEM
EPSON FX1170 A3 9 pina (300 cps) YU, kabl =1050DEM
EPSON LQ100 A4 24 pina (200 cps) YU,kabl=580DEM
EPSON LQ570+ A4 24 pina (269 cps) YU, kabl =820DEM
EPSON LQ1070 A3 24 pina (225 cps) , kabl =1180DEM

HP IV L,1 MB, (300 dpi), kabl =1850DEM
HP IV,2MB, (600 dpi), kabl =3450DEM
Streamer IRVIN 120Mb=480DEM
Streamer IRVIN 250Mb=580DEM
MODEM externi MNP5 V 42 bis=350DEM
FAX-MODEM int. MNP5 V 42 bis=200DEM
Handy scaner =330DEM

. . . a naše prednosti su:

2 GODINE GARANCIJE

ISPORUKA NA ADRESU KUPCA

DILERSKI POPUST 3-10%

[illegible]MHz

14" SVGA mono monitor
RAM 2Mb
HDD 42 Mb
FDD 5.25" ili 3.5"
SVGA card 512 kb
AT bus I/O kontroler 2s/1p/1g
mini tower+tastatura+mouse

=1190DEM

386DX/40MHz

14" SVGA mono monitor
RAM 4Mb+128Kb cache
HDD 125 Mb
FDD 5.25" ili 3.5"
SVGA card 512 kb
AT bus I/O kontroler 2s/1p/1g
mini tower+tastatura+mouse

=1600DEM

486D[illegible] 33M[illegible]

14" SVGA mono monitor
RAM 4Mb+256Kb cache
HDD 170 Mb
FDD 5.25"+ 3.5"
SVGA card 512 kb
AT bus I/O kontroler 2s/1p/1g
midi tower+tastatura+mouse

=2250DEM

486DX/50MHz

14" SVGA mono monitor
RAM 4Mb+256Kb cache
HDD 170 Mb
FDD 5.25"+ 3.5"
SVGA card 512 kb
AT bus I/O kontroler 2s/1p/1g
midi tower+tastatura+mouse

=2550DEM

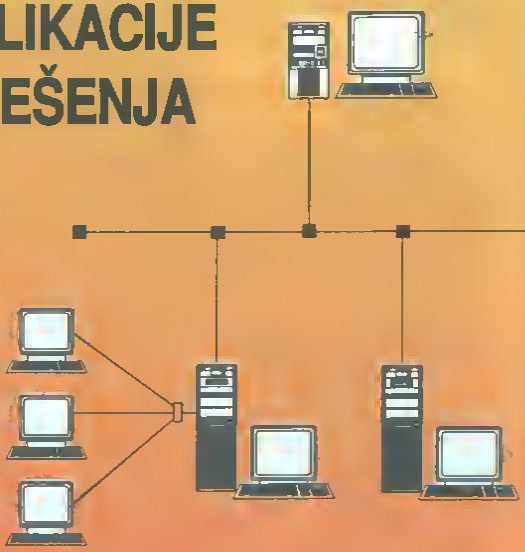
[illegible]

14" SVGA mono monitor
RAM 4Mb+256Kb cache
HDD 170 Mb
FDD 5.25"+ 3.5"
SVGA card 512 kb
AT bus I/O kontroler 2s/1p/1g
midi tower+tastatura+mouse

=2800DEM

MI OVO RADIMO DUŽE OD VAS

PC&C *Personal Computer & Communication*
U N I X
NETWORKING
RELATIONAL DATABASE
POSLOVNE APLIKACIJE
KOMPLETNA REŠENJA
i . . .

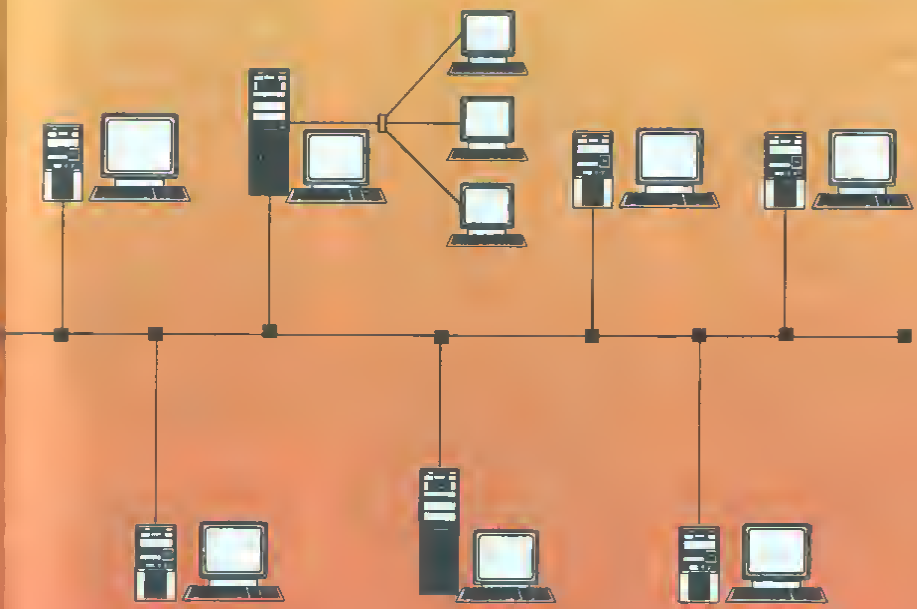


MS SUPPORT

Novi Sad, Beočin
021-611-366, 52-550

MICROSYS

10 GODINA + 1 GODINA + ••• GODINA



Beograd, Molerova 70
011-432-690, 430-059

ZA LIČNU UPOTREBU

U skladu sa tradicijom, Hewlett Packard nastavlja da varira osnovna modele svojih laserskih štampača da bi ih što bolja prilagodio različitim potrebama na tržištu. Nedugo nakon apsolutnog iznenađenja sa modelom 4L, koji je namenjen praktično za svaku kuću u kojoj postoji računar, na tržištu se pojavio i štampač za ličnu upotrebu sa oznakom 4P. Njemu je, sa modela 4, oduzeto samo ono bez čega se može, i dodato, sa modala 4L, nekoliko zanimljivih opcija. Rezultat ovih tehnoloških zahvata je varovatno najjeftiniji laserski štampač sa rezolucijom 600x600 na avetu.

Jovan Regesok

Štampač se oznakom LaserJet 4P stigao je u redakciju u običnoj – plastičnoj kesi. Ovaj bizaran detalj svakako na bismo pomenuli kad se iz njega na bi mogao izvući i jedan zaključak od interesa se ovaj test: da je u pitanju devetlino lagan (9 kg bez tonera) i lakši|iv model da podnese tako nehajan transport. U pitanju je poslednji izdanak u četvrtoj generaciji Hewlett Packard laserskih štampača namenjen, pre svega, pojedinačnom korisniku, koji drži do visokog kvaliteta štampe, ali ga na koncu prevše interesuje da bi mu bile važne tako skupe opcije kao što su robusnost i velika brzina rada.

SAMO MALO DRUKČIJA

Personalni model HP 4P je izveden iz profesionalnog modela LaserJet 4 i ima sve njegove ključne osobine – radi u rezoluciji 600x600 tačaka po inču se RET tehnologijom za korekciju finoće ivica i tonerom se veoma finim zrnom (*microfine*), se računarom se povezuje preko veoma brzog, *Bi*7*rectcs* paralelnog dvosmernog interfejsa, ima 45 ugrađenih skalabilnih fontova. Štampač je, u skladu se svojom osnovnom namenom, pojednostavljen samo u onim osobinama koje nisu od naročitog značaja se pojedinačnu upotrebu ima sporiji procesor i sporiju mašinu (štampa četiri umesto osam strana u minuti) i nešto skromnije mogućnosti proširivanja – broj slotova se proširenje memorije je smanjen se četiri na tri, a mogućnost priključivanja dodatnih magacina se papir i modulemi I/O port su ukinuti.

Na verujemo da će se neko, u kućnim uslovima, osetiti prikraćen što ne može odjednom da ubaci 500 umesto 250 listova papira, a što mora ručno da ubacuje koverte u štampač umesto da ih on uzima automatski iz specijalnog magacina. Modularni I/O port, međutim, služi se vesu štampača se svetom – on je, pre svega, predviđen se mrežne kartice, all ga neki proizvođači, poput firme LaserMaster, koriste se unapređivanja performansi štampača – i mi bismo se, da smo kojim slučajem vlasnik jedne ovakve mašine, prijatnije osećali da je taj port ipak zadržan.

U štampač HP 4P je ugrađen 32-bitni RISC procesor i960 na 16 MHz se 2 MB memorije. Specijalna tehnika se štednju memorije komprimovanjem slika i fontove (poznata po akronimu MET *Memory Enhancement Technology*, i prvi put primenjena na modelu 4L) gotovo udvostručuje ovaj kapacitet. Hewlett Packard smatra da je to dovoljno se štampanje teksta i grafike u najvećoj rasolucij, all i primaje, konačno, da vam u pojedinim slučajevima, kod izuzetno složenih strana, može ustrebati i neki megabajt više – na primer od 4 do 26 MB, na koliko se maksimalno može proširi memorija preko tri slota se proširenja. „Složena strana" je relativan pojam, koga postajete svesni onog trenutka kada vam štampač, zbog nedostatka memorije, podeli stranicu na dva dela. Model na testu je imao 6 MB i mi smo bili pošteđeni bilo kakvih neprijatnih iznenađenja. Sa modela 4L prenete je još jedna korisna osobina – *EconoMode* se štednju tonera se 50%.

Model HP4P se, kao i ostali štampači u seriji 4, izrađuje u dve varijante. Prve, se oznakom P je namenjena PC kompatibilnima nešto skromnijih potreba, a druga, se oznakom PM, pre svega vlasnicima „mekintoš" računara i naprednijim korisnicima Windowe-a. U nju je ugrađen nešto brži i960 RISC (20 MHz), AppleTalk interfejs, 6 MB RAM-a i, poslednje i najvažnije, PostScript interpreter, bez koga se na može ni zamisliti primena štampača u bilo ozbiljnijim grafičkim poslovima. Na PostScript se, preko jednog od tri slota za proširenja, uz obaveznu proširivanjem memorije na 6 MB, može dogradi i model P, tako mislimo da je bolje odmah procenili prave potrebe i u startu nabavili 4PM

NA SVEMU I SVAČEMU

Nakon kratkog modernističkog izleta se modelom HP 4/4M. HP dizajeri su se ponovo vratili klasičnom dizajnu – kontrolni panel se displejom je na prednjoj strani, a ranovni obrisi štampača više podsećaju na skraćenu dvojicu (HP Series II+), nego na stanjenu četvorku. Za tri osnovna modela u familiji 4 zaista bi malo ko rekao da, po izgledu, pripadaju istoj porodici all su zato u teanom srodstvu po svojim (dobrim) osobinama. Štampač je kompaktan, zauzima malo mesta na stolu i lzuzetno je tih i čist. U stanju mirovanja (na ješost, nije u pitanju instant tehnologija iz modela 4L) ga jedva možete čuti kako diše, a 5W energije koliko troši se održavanje minimalnih funkcija, neće povećati temperaturu u prostoriji ni se delić stepena (HP 4/4M dok miruje vuče 90 W, a pod opterećenjem i čitavih 800 Wt). Personalna četvorka je proizvedena u skladu se „EPA Energy Star" standardima se uređaje koji na zagađuju životnu sredinu Mi smo u toku teste štampač držali bliže nego na dohvatu ruke, a mirne duše bismo ga stavili i na sto se koga jedamo.

Laserski štampači, po tradiciji, prihvataju različite formate i vrste materijala. Magacin se papir, koji se poput fioka, izvlači iz štampača, prima do 250 listova papira u formatu od executive (184x267mm), preko *letter* (216x27ll) i *A4* (210x297), do *legal* (216x356mm) Magacin je univerzalnog tipa, a na potrebni format se podešava preko odgovarajućih graničnika. Spisak nalepnica, fnlja, koverta i papira nestandardnih formate (najmanji format je 76x127mm), koji se ubacuju u štampač isključivo ručno, toliko je begat da je proizvođač osetio potrebu da ovaj model nazove *Custom Media* (što će reći da štampa na svemu što vam može pasti na pamet – pod uslovom da je savitljivo, da nije deblje od 0 16 mm i da se na topi do 200° C se manje od 0,1 sekunde) Kapacitet od 250 listova je, se teoretskom brzinom od 4 stranice u minuti, dovoljan se sat vremena nepreklnog štampanja, što znači da vas ovaj model neće zamarati potrebom da ga stalno opsluživate. Uostalom, i-stovetan kapacitet ima, bez dodatnih magacina, i model HP4, koji je predviđen se mnogo intenzivniju upotrebu u kancelarijama i mrežama. (Na ovaj model je, inače, moguće priključiti i univerzalni magacin kapaciteta 500 listova, kao i magacin se koverta kapaciteta 75 listova.)

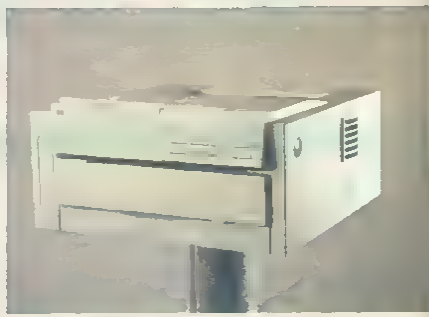
UNIVERZALNI DRAJVER

Uz HP 4P dobiate i četiri diskete se već tradicionalnim HP softverom – drajverima se DOS i *Windows* svet i programima HP *Explorer* (originalni vodič kroz osobine i opcije štampača) i *Remote Control Panel* (izbor opcija softvenski iz računara, umesto hardverski, se kontrolnog panela) Na *Windowe* disketama je štampan poznati *Windowe* logo, a među ključnim opcijama štampača, na koje i sam proizvođač skreće posebnu pažnju, nalazi se i „lzuzetno štampanje iz *Windows-a*" (*Great Windows Printing*) Iz toga bi se mogio zaključiti da se Hewlett Packard ovoga puta, se razliku od dosadašnje prakse, potrudio malo više da obezbedi potpunu softversku podršku za neke inače izuzetne all lelko dostupne opcije svog sopstvenog proizvoda. Na žalost, u čitavih pet i po megabajta, na koliko se softver protaže u punoj instalaciji pod *Windows*-ima, nije se našao ni jedan jedini bajt prostora se direktnu podršku različitim kodnim stranama, i se tog stanovišta novi HP drajveri još nisu zaslužili da ponesu *Windowe* logo – znak programa koji pokriuje izuzetnu pažnju primeni u raznim romanskim, germanskim i slovenskim jezicima.

Najnovija verzija HPPCL5e drajvera, se oznakom 31 V1.25 uzima na disku oko 1,5 MB prostora i donosi nekoliko prijatnih novina. Obuhvate podršku se sve modele iz familje 4 (4/4M, 4L/4ML, 4P/4PM I 4Si/4Si MX), pri čemu se odgovarajuća verzija generiše u procesu instalacije, a preko odgovarajućih dijaloga u **Printer Setup** sekciji može se prelupati prilično broju opcija štampača – gršći kvalitet (rezolucija, gustina štampe, ekonomični režim, RET opcije, podrška se HP-GL/2) i podela unutrašnjih resursa. Na vidimo, međutim zašto preko ove sekcije nisu dostupne i sve ostale opcije kojima se upravlja iz programa *Remote Control Panel*, kao što je na primer, izbor osnovnog panta i njegove kodne strane. Na vidimo, zapravo, zašto Hewlett Packard na bi dozvolio i omogućio potpunu softversku kontrolu štampača, dakle svih opcija kojima se prelupa preko kontrolnog panela, kao kod modela 4L? Za sada, softver u računaru hramije za softverom u štampaču.

NEŠTO SU OPET ZABORAVILI

Nijedna osobina laserskog štampača nije se potcenjivanje all je ključno samo jedno – fontovi i grafika. Kvalitet štampe Hewlett Packard laserskih štampača je oduvek bio dostojan svakog poštovanja. Sa svojih



KORISNIKU SE NE MOŽE UGODITI

Još je Reasonjin Hoclita znao da se svetu ne može ugoditi. Mi tome dodali – konkurentima računara pogotovu. I pored svih mogućnosti koje im proizvođači softvera i hardvera stavljaju na raspolaganje se prilagođavanje njihovih proizvoda specifičnim potrebama, još se, očigledno, tu i tamo nešto prilagođava i neispravlja sredstvima, koja čine deo preinstaliranog softvera.

U prilazu štampaču 4L dali smo nekoliko ideja kako se štampač može prevariti da radi se bilo kojom stranom. Preporučili smo da se u modulu HPPCL3e.dll kontrolni string „19U" kojim se postiva kodna strana *Latin 1*, prepravi u string „09E" kojim se postiva kodna strana *Windowse 3.1 Latin 2*, dakle ona koja je usklađena sa verzijom za istočnoevropska jezika. Tom trivijalnom operacijom neke se usklađuju slova u štampaču sa slovima na tastaturi i ekranu. Ako se ta zamena izvrši na dvanaest pre instalacije, onda se obezbeđuje samo na jednom mestu, a ako se vrši na bilo kom drugom nekom instalacije, na ovoliko mesta koliko ima fontova u štampaču. Ova intervencija je imala i jednu sporednu manu – se ovako prepravljanim drajverom nije moguća ponovnosti desnu marginu, jer *Windows* i dalje uzima podatke o širinama slova iz datoteke *Windowse Latin 1*.

Zahvaljujući bibliotekama *TrueType* fontova, koja se isporučuju uz verziju V1.25 HPPCL5E drajvera, štampač se može prevariti se pet-pet puta. Dovoljno je neka slova se poklapa na jednom mestu, a ako se vrši na bilo kom drugom nekom instalacije, na on

ČIKAGO BEZ MASKE

Welcome to the Chicago Beta Program.

Microsoft već uveliko pripreme naslednika sadašnjeg *Windows*-a. On se krije iza šifre „Chicago" i trenutno se nalazi u fazi beta-testiranja, a ne tržištu će se pojaviti krajem 1994. godine. Otkrivamo najizvesnije novina koje će doneti *Windows 4.0*.

Svima je poznato da Microsoft već godinu i po dana priprema dva velika operativna sistema. Onaj po imenu Cairo će naslediti sve veze sa prošlošću – to je potpuno objektno orijentisani sistem, koji će postepeno zameniti današnji *Windows NT*. Drugi poznat kao Chicago je privrženiji tradiciji i treba da zameni današnju *Windows*/DOS kombinaciju, a biće predstavljen krajem 1994. godine kao *Windows 4.0*.

Chicago je bio predmet raznih nagađanja. Po jednima, to je sistem se potpuno zaštićenim režimom rada, kao OS/2 i *Windows NT*, i se mora da se oslanja se MS-DOS. Drugi kažu da ima objektni desktop interfejs, koji treba da poslužiti kao stepenik se prelazak se Cairo. Neki su, opet, pričali da će to biti *NT* Lite, koji će pružati poumljano multi-tasking okruženje za intenzivan i obiman rad, ali bez sigurnosti i podrške se mrežni server, koju nudi *Windows NT*

Beta-verzija daje odgovore na mnoga od ovih pitanja, mada Microsoft napominje da ona se odnašava o potpunosti osobine novog sistema – one bi se nalme, do konačne verzije mogle radikalno promeniti. I-pak, se prvom testiranju softver se pokazao mnogo pouzdanijim i dovršenijim nego što bi se očekivalo od jednog prototipa. Dakle, ono što vidite verovatno **jeste** *Windows 4.0*

ISTE STVARI

U osnovi, Chicago nudi iste stvari koje i sadašnji Windows: ima grafički desktop se kojim možete organizovati datoteke, kontrolisati štampač, pokretati DOS, kao i prave *Windows* aplikacije. Podržava sve standardne *Windows* funkcije kao što su clipboard, OLE i DDE.

Međutim, način na koji su ove opcije predstavljene je sasvim nov. Naime, Chicago ima sasvim novi *front-end shell* koji se zove **File Cabinet** i prevazilazi *Windows* **Program Manager** i **File Manager** integrišući upravljanje datotekama i programima u jednoj grupi prozora se desktop folders (mnogo više podseća se Apple Macintosh ili OS/2 *Workplace Shell* nego se *Windows* interfejs)

Na prvi pogled, Chicago arhitektura je iznenađujuće konvencionalna i čak se učitava preko DOS-a, kao i današnji sistemi. Međutim Chicago je sposoban da izvršava 32-bitne *Windows NT* aplikacije, što ukazuje se značajnu promenu u unutrašnjoj strukturi – Chicago ima potpuno 32-bitno jezgro. Još jedna dobrodošla novina je prihvatanje dugačkih imena datoteka koje se postiže putem standardnog DOS FAT (*File Allocation Table*) diek sistema, a se naprednijeg NTFS (*NT Filing System*), što znači da je održana kompatibilnost se ranijim DOS i *Windows* verzijama.

SISTEMSKA ARHITEKTURA

Suprotno govorkanjima, Chicago se (barem u ovoj verziji) nije oslobodio zavisnosti od DOS-a, mada donekle stvara iluziju da jeste. Pri podizanju sistema, ukazuje se Chicago logotip pre nego što sistem uđe u *Windows 4.0* desktop. Ali je to samo fasada – ukoliko uklonite datoteku koje sadrži logotip, videćete kako DOS učitava drajvere uređaja i rezidentne programe se dobro poznati stari način. Sistem je podešen da automatski pokreće WIN.COM. Ali ukoliko uklonite taj patch dobićete običnl C prompt. Ako pak uklonite HIMEM.SYS ili datoteke CONFIG.SYS, *Windows 3.1*, što dokazuje da se radi o lažnom stvarnom režimu. Ipak, podržava datoteke se dugačkim nazivima.

Naravno, može se desiti da ovo oslanjanje se DOS ličšeno do konačne verzije. Ali čak i ako se to se desi, se znači da *Windows 4.0* neće biti operativni sistem se potpuno zaštićenim režimom rada. Sa svakom novom verzijom *Windows* je preuzimao sve veći broj sistemskih funkcija od DOS-a (npr. novi *Windows for Workgroups 3.11* preuzima se sebe upravljanje FAT om), tako da se njegova uloga svela se puko učitavanje. Sasvim je logično što Microsoft se ovaj način koristi DOS, pošto *Windows* sistemi svakako moraju sadržati DOS kod da bi mogli podržavati DOS aplikacije. Postojanje DOS sistema u pozadini nudi trenutni povratak u *Windows 3.1*, što je vrlo praktično ukoliko se javi bilo kakav problem.

Šta donosi Chicago?

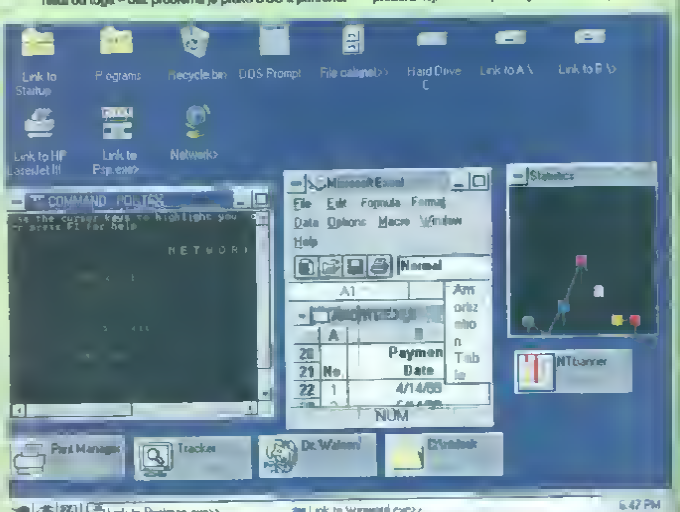
- File Cabinet, novi desktop interfejs
- Izvršava se nad operativnim sistemom MS-DOS 7.0
- Podržava duga imena datoteke
- Izvršava 32-bitne *Windows NT* programe
- Mrežne funkcije kao standard
- Multitasking, ali se *pre-emptive*

slowe neće moći da se pokrene, što potvrđuje da je to još uvek samo DOS ekstender koji očekuje da iz 386 zaštićenog režima pređe u DOS-ov stvarni režim da bi mogao obavljati hardverske servise.

DOS nad kojim se Chicago izvršava je nazvan Chicago 4.0, ali Microsoft-ov dijagnostički program MSD otkriva njegovo interno ime – MS-DOS 7.0 i o njemu su iz raznih izvora stigla neproverena veoti da će nuditi zaštićeni režim rada i multitasking. Međutim, ništa od toga – bez problema je preko DOS-a pokrenut

CHICAGO DESKTOP

Najveća promena je desktop interfejs, koji se zove **File Cabinet**. Microsoft se konačno oslobodio inhibicija (izazvanih sudskim sporenjem) zbog kojih je oklevao da prozore es foldere (podkataloge) prikaže direktno se desktopu (kao kod Macintosha). Nema više **Program Manager**-a, njegove opcije **Program Groups** kao ni **File Manager**-a. Umesto njih, tu je skup prozora koji direktno počivaju se strukturi podkataloga.



Folder (podkatalog) ekvivalent je simbolan direktorijuma na Chicago desktopu. Folderi se otvaraju slično kao na Macintosh ili OS/2 sistemima.

Minimizirane ikone za razlikuju se sasvim novom stilu. I prozori se aplikacije i prozori se foldera se smanjuju „laskavima" i mogu se minimizirati. Nema minimizirnog task-prostora kao kod OS/2.

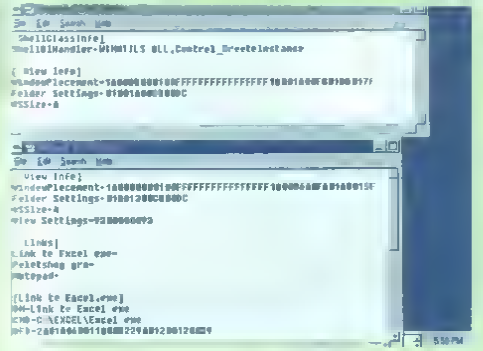
Recycling bin omogućava drag-and-drop postupak za brisanje datoteka, kao kao Mac sistema. Datoteke se mogu povratiti iz *Bin*-a.

DOS, Windows i *Windows NT* aplikacije se izvršavaju zajedno na desktopu.

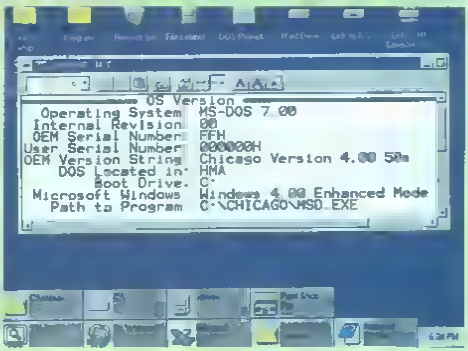
Tray je specijalni folder čiji je sadržaj dostupan iz svakom trenutku.

Link, ili veza se direktorijum. *File Cabinet* nudi opciju kreiranje koja otvara prozor foldera i omogućava beskonačno datoteka i fajlova svojstava (properties)

CHICAGO: IZA INTERFEJSA



DESKTOP.INI datoteke, smeštene u svaki folder, sadrže informacije o vezama i specijalnim funkcijama. U folderu CONTROL.SYS (gore) jedna od datoteka poziva specijalni DLL koji vrši *Control Panel* funkciju. Dole je DESKTOP.INI se vezama se datoteke u drugim folderima.



Microsoft-ov MSD program se dijagnostiku otkriva ratinu o DOS-u 7.0 i Windows-u 4.0. Sve je jasno kada se zna da je DOS smešten u HMA i da se Windows 4.0 izvršava u enhanced režimu.

ja na tvrdom disku i u kojima se obavlja baratanje datotekama i učitavanje programa kao i na Mac II OS/2 sistemu.

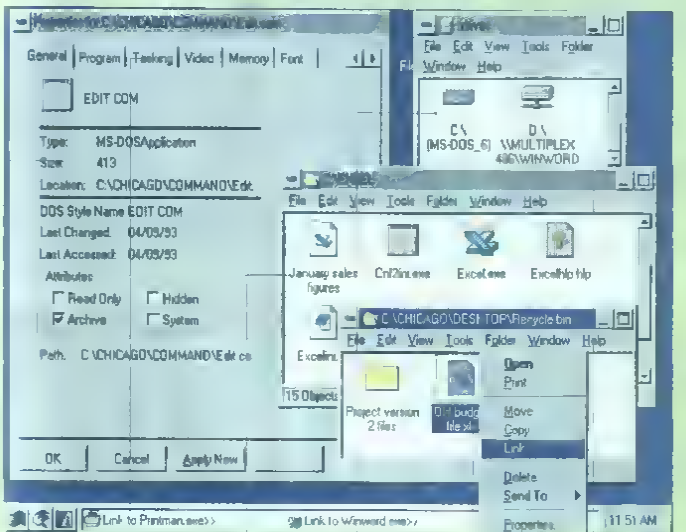
U osnovi, *Chicago* desktop je poput velikog **File Manager-a** – program se pokreće dvostrukim klikom na njegovu ikonicu; ukoliko na ikonu programa dovučete neku datoteku-dokument, učitaće se zajedno, a kada datoteku dovučete iz jednog foldera u drugi, ona se premešta ili kopira. Za razliku od **File Manager-a**, možete pomerati objekte na sam desktop koji je, u stvari, samo poddirektorij po imenu „DESKTOP". Prozori za aplikacije i foldere se ponašaju ugnavnom na isti način, tako da jedan *spreadsheet* i podatke za njega možete imati jedno do drugog, kao minimizirane ikonice u dnu ekrana. Editovanje dokumenata se obavlja pomoću *drag-and-drop* opcije, što je zgodnije nego kada su dostupni samo preko posebne **File Manager** aplikacije

Microsoft je se svoj *Chicago* desktop posudio mnoge funkcije iz sistema OS/2, gde se on zove **Workplace Shell**. Na primer desni klik na neki objekat (ikona datoteke) otvara *pop-up* meni koji nudi opcije **move, copy** i **delete**, kao i opciju **properties**, koja otvara beležnicu (*notebook*) sa tabovima i vrlo listova – poput opcije *Setting* book kod sistema OS/2. Ipak, *Chicago* se nije tako strogo držao objektne strane kao OS/2 – *Chicago* folder prozori imaju tastarske menije, kao i **File Manager** i nude ste funkcije kao i *pop-up* meniji ali u tradicionalnom obliku

Današnje programske grupe u **Windowse-u** služe sa grupisanje referenci na određene aplikacije koje su smeštene u različitim poddirektorijima. Sada ovo obavlja opcija zvana **Links**. Može se upotrebiti sa opcijama *Alias* kod Maca ili *Shadow* kod OS/2 – predstavlja referencu ili vezu sa datotekom koja je smeštena na drugom mestu. Programska grupa iz *Windows*-a 3.1 (GRP) je zamenjena folderom punim raznih veza (*links*) – kada instalirate *Chicago* stare GRP datoteke se automatski konvertuju. Međutim s obzirom da se **Link** može odnositi na bilo koji objekat možete kreirati i grupe dokumenata, ili mešati podatke i programske datoteke u jednoj grupi **Links** možete smestiti na desktop.

Ove „veze" se ne čuvaju u obliku datoteke, već se nalaze u skrivenoj DESKTOP.INI datoteci koja se nalazi u svakom folderu. DESKTOP.INI datoteke može da sadrži, osim linkova i informacija o veličini prozora, poziva se DLL rutine koje menjaju ponašanje sistema kada se na desktopu otvori prozor određenog foldera. Dobar primer je CONTROL.SYS folder koji zamenjuje **Control Panel** iz *Windows*-a 3.x. Kada otvorite CONTROL.SYS poziva se određeni DLL iz DESKTOP.INI datoteke i generiše skup ikona se miša, boja itd. kao i kod stare slatke.

Ovaj sistem se izdalno koristi u celom *Chicago* paketu a može se i proširiti – svaki proizvođač softvera može ponuditi neki svoj DLL i poslužiti ga iz datoteke DESKTOP.INI Tako, na primer možete imati Lotus folder koji nudi specijalni meni zasnovan na tastaturama kada ga otvorite. Savršen način da izmaš starog DOS sistema



Selektira se upisivanje serverskih svojstava datoteke, u OS/2 sistemu. Sadržaj datoteke se menja u zavisnosti od tipa računara.

Stranice se tabelarno omogućavaju da se se desktopa direktno kontroliše *runtime* opcije određenog DOS programa.

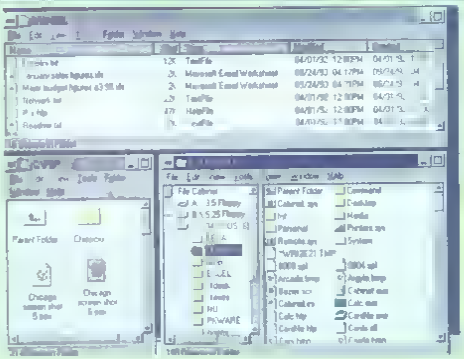
Prozor se folder koji otvara *File Cabinet Explorer* Datoteke se kopiraju i premeštaju pravljenjem iz jednog prozora u drugi.

Explorer root prozor u kojem se vidi raspoređivi prostor na disku (sastoji i mrežni drajv)

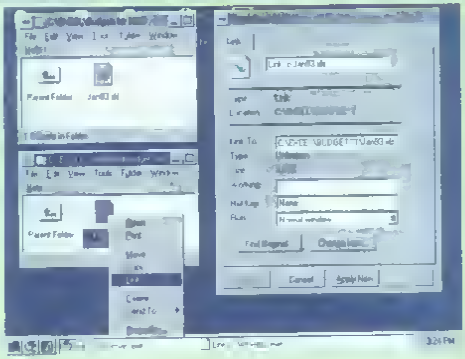
Pop-up meni koji se dobija kada se desnim tasterom miša klikne na ikonu željenog objekta.

Windows 4.0 ima osvrin novu naslovnu liniju i ikone s kojima ih sve aplikacije.

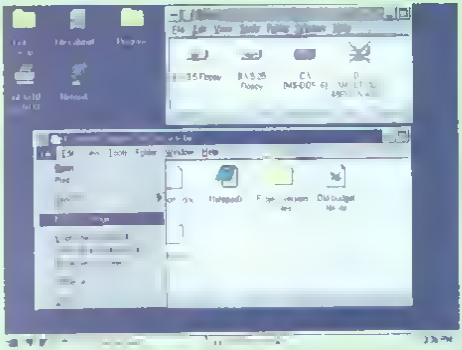
UPRAVLJANJE DATOTEKAMA NA DESKTOPU



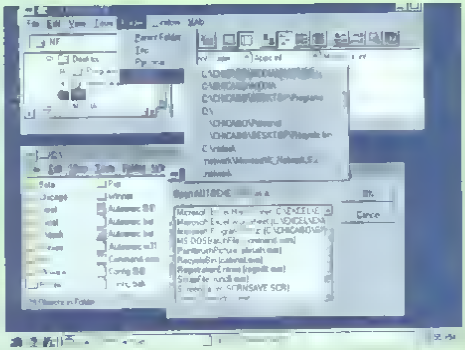
Prikazi za foldere se mogu formatirati na razne načine. Lista u gornjem delu ukazuje na pokušaj da se imitira Mac način davanja informacija o tipu datoteke, ali je on još uvek nedovršen. Sa dugačkim imenima datoteka (dole levo) trenutno se bolje nosi desktop nego DOS 7.0. Stablo (desno) podseća na *File Manager*.



Links zamenjuju neke opcije iz starog *Program Manager*-a i predstavljaju vezu sa datotekama i sa programima. Kontrola objekata se postiže preko beležnica u koje se zapisane njihova svojstva (desno), a prikaz im se preko menija koji se dobija klikom desnog tastera, ili iz već tradicionalnih menija koji se nalaze u prozorima foldera.



Reciklirani bin je običan folder (RECYCLE.BIN), ali *File Cabinet* automatski dodaje *Nuke* i druge opcije njegovom standardnom meniju. Neobično je da se njihovo arhiviranje ne koristi DESKTOP.INI DLL pozive, za razliku od ostalih foldera, kao što je CONTROL.SYS.



(Gore) Desktop je preuređen mnoštvom sitnih pomagala – svaki nudi listu prethodno posećenih foldera u koje se začas možete vratiti. (Dole) *Chicago* kvalitetno nalazi asocijacije sa datoteke DOS tipa i nudi izbor aplikacija se njihovo editovanje.

datoteka postavite aktivne objekte za skladištenje.

Chicago desktop arhitektura se zaokružuje sa nekoliko posebnih dodataka. Neki od njih kao **Network** objekte se „virtuelni folderi" koji predstavljaju sistemske resurse u standardnom formatu foldera i ikona. Drugi pak, kao **Tary**, su pravi folderi se specijalnom namenom – datoteke i veze se u njemu prikazuju u posebnom prozoru, **System Control**, koji je uvek prisutan na desktopu i za sada ima tri tastera se uključivanje taskova, lociranje objekata i pomoć u radu se sistemom.

Korisnici koje plaši sasvim novi interfejs ne treba da brinu. Tu su sve stare **Program Manager** i **File Manager** aplikacije i mogu se pokrenuti se DESKTOP-a. Čak se kao „shell" može odabrati **Program Manager** kada se dobija stari *Windows* desktop bez **File Cabinet**-a.

DUGAČKA IMENA DATOTEKA U DOS-u I WINDOWS-u

Ograničene dužina imena datoteka jedan je od retkih nedostataka DOS-a i *Windows*-a. Jednostavno ste primorani da datoteku nazovete JANPRSAL.DOC e u stvari biste voleli da je nazovete „Prodajni saldo za januar" *Chicago* će pokušati da ovaj nedostatak ispravi, primenjujući metod za podršku dugačkih imena, koji je ipak kompatibilan se postojećim DOS diskovima.

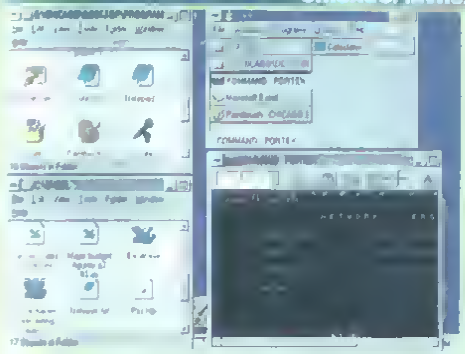
Kod sistema *Chicago*, svaka datoteka ima standardno ime u obliku FILENAME.EXT koje se pohranjuje u određeni direktorij na disku na uobičajen način. No, datoteke može imati i dodatno ime dužine do 256 karaktera, uključujući i razmak, koje se takođe čuva u DOS direktoriju i koristi slotove koji fizički prethode „stvarnom" upisu datoteke

I DOS 7.0 i *Chicago File Cabinet* će umeti da se nose sa dugačkim imenima, mada standardne DOS komande zahtevaju da prvih osam karaktera dugačkog imena budu jedinstveni – kod sličnih imena se javljaju problemi. Kada generišete neko dugačko ime sistem automatski kreira „pravo" kratko ime

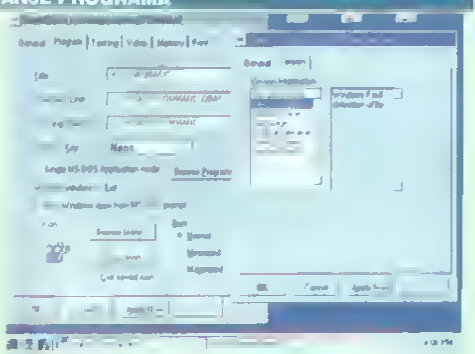
Međutim, se sada se problem aplikacije – ni jedna od testiranih nije prihvatala dugačke nazive datoteka. Ovo se može ispraviti novim sistemskim kodom, e nadajmo se da će se to i desiti pre izlaska konačne verzije.

Ekstenzije u imenu datoteka (.XLS, TXT) još uvek se koriste kao znak raspoznavanja – da se vidi kojoj datoteka pripada kojoj aplikaciji. Kod Apple Macintosh sistema sam upis u direktorij nedvosmisleno vezuje datoteku sa aplikaciju u kojoj je nastala, dok *Chicago* u tome još uvek zaostaje.

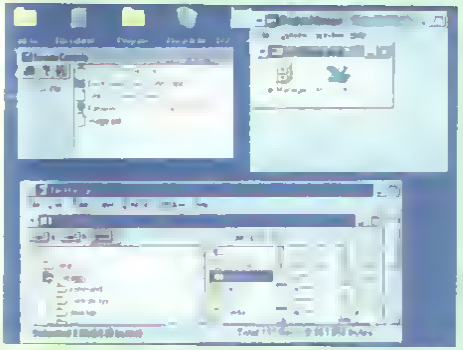
CHICAGO: IZVRŠAVANJE PROGRAMA



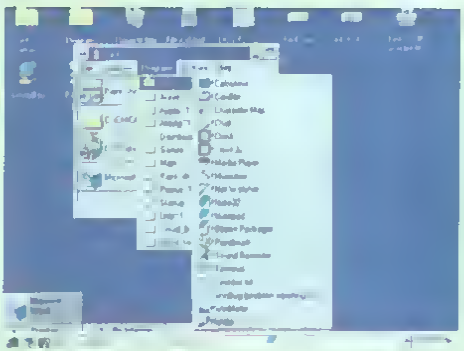
Nova lista taskova (gore desno) prikazuje prozore foldera kao i programe, a naglašeno je integrisano upravljanje datotekama i programima. DOS prozori (dole) mogu imati tasterski meni koji nudi trenutan pristup fontovima i tasking opcijama. Programske grupe, kao što je „Accessories" (gore levo) sada se zamenjene folderima koji se puni veze se raznim aplikacijama.



Sadržaj PIF-a (*Program Information Files*) DOS aplikacije se može dobiti u obliku stranice iz beležnice (*notebook*), a DOS prozor ima tasterski meni se trenutno menjanje fontova. Za *Windows* aplikacije (desno) se date informacije o registraciji. NT aplikacije se tretiraju kao DOS (nepoznate) datoteke, ali pozivaju 32-bitni podsistem kada se učitaju.



Tray („posluživnik", gore levo) služi za brz pristup često korišćenim objektima (kao *Shelf* u paketu *NeXTSTEP*). *Program Manager* iz *Windows*-a 3 se se novog desktopa može pokrenuti kao task.



Chicago direktorij kao i foldere se specijalnim funkcijama – npr. PROGRAMS, čiji se sadržaj i poddirektoriji dobijaju preko redizajniranog prozora *Task List*.

POKRETANJE APLIKACIJA

Dobra vest je – ovde nema promena. Međutim, to je istovremeno i loša vest, jer znači da nema ništa od zakazivanog *pre-emptive* multitaskinga o kojem se najviše nagađalo. Drugim rečima, svaka „nevaspitana" *Windows* aplikacija može da prigrabi mašinu i zakoči sve ostale programe. Što se i potvrdilo pri startovanju jednostavnog *Visual Basic* programa se zatvorenom petljom. Ovo je verovatno oblast koja će morati da pretrpi promene pre konačnog izdanja.

Windows aplikacije se izvršavaju istovetno kao ranije, mada izgledaju nešto drukčije zbog novog dizajna menija i dijalog-prozora. DOS aplikacije se mogu odvisti na celom ekranu ili u prozoru kao i do sada – u drugom slučaju korisnik ima na raspolaganju novi tasterski meni (*toolbar*) se izbor fontova i drugih opcija. Testiranje nije pokazalo nikakve probleme u kompatibilnosti.

Veliko iznenađenje je mogućnost da se direktno izvršavaju 32-bitne *Windows NT* aplikacije, uporedo se postojećim DOS i *Windows* programima. Radi se o značajnom zaokretu u arhitekturi jezgro samog *Windows*-a je sada 32-bitno, se podrškom za novi Win32C API (C se odnosi na *Chicago*), koji sadrži većinu NT karakteristika, uključujući obradu više niti (*multithreading*), klijent verzije imenovanih pajpova, poziva udaljenih (*remote*) procedura – izostavljene se samo najnaprednije funkcije koje se tiču grafike sigurnosti i servera.

Dakle, situacija je obrnuta u odnosu na sadašnji *Windows* – kod Chicaga je Win32C cenovni (*native*) API, dok se preko posebnih kodnih modula podržava ju 16-bitne aplikacije.

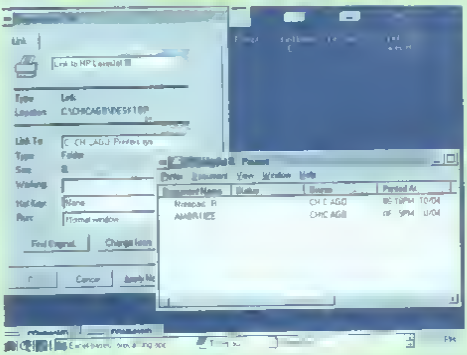
NT aplikacije se izvršavaju u vlastitoj mašini po imenu **Virtual Machine** (VM), koja je odvojena od 16-bitne *Windows* mašine. Kada je prilikom testiranje pokrenuta jedna 16-bitna C++ aplikacija (koja je zakočila celu VM *Windows* mašinu), NT aplikacije se nastavile da ažuriraju svoj prikaz, ali se im meniji bili „zaključani". Što pokazuje da i desktop koristi ali VM. Ovo nije nivo robusnosti koji pruža NT i dobro bi bilo da se u tom pogledo nešto promeni do konačne verzije.

MREŽNE FUNKCIJE I KONTROLA

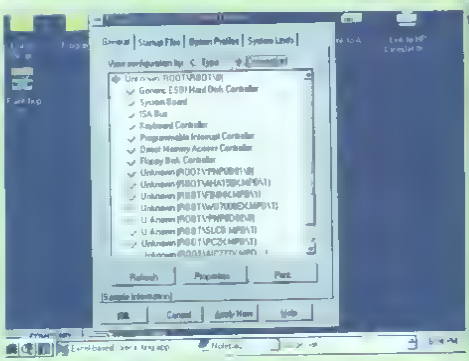
Microsoft je već najavio da se *Windows* i *Windows for Workgroups* više neće prodavati zasebno, te da će podrška se *peer-to-peer* mreže i grupni rad postati standard za sve njihove operativne sisteme. Beta-verzija softvera *Chicago* ima ugrađen program *Windows for Workgroups 3.11*, ali se nekoliko zanimljivih varijacija i dodataka.

Korištenje istih datoteka se kontroliše preko menija iz foldera **File Cabinet**, dok je za zajedničku upotrebu istog štampača još uvek potreban stari **Print Manager** program. Inače, trebalo bi da se u konačnoj verziji sav *sharing* (deljenje resursa) kontroliše dvostrukim klikom na odgovarajuću ikonu. *Chicago* nudi i neke nove (i nedovršene) funkcije za *browsing* (pregled) u mreži, mada je povezivanje se udaljenim drajverom se sada teže nego u *Windows*-u 3.11. Mrežni drajvovi

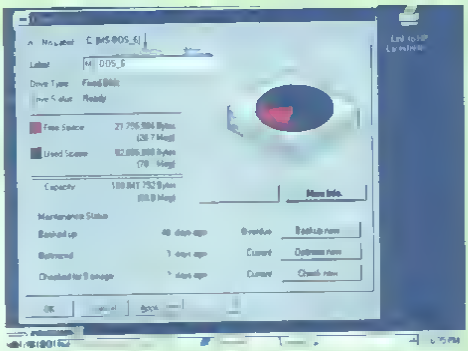
CHICAGO: KONTROLA SISTEMA



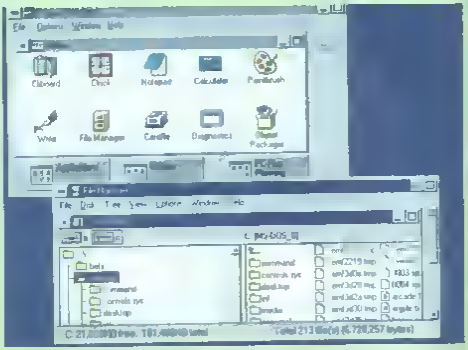
Mrežne sposobnosti se zasnovane na programu *Windows for Workgroups* koji se očekuje uskoro. Zajedničko korišćenje istog štampača se postiže preko *Print Manager-a*, ali će se kasnije postizati preko beležnice samog objekta za štampač.



Instalacija CD-ROM drajva i mnogih drugih perifernih uređaja će biti (relativno) lake, uz pomoć *plug-and-play* drajvera uređaja i ovog konfiguracionog pruzora. Biće moguće menjanje rezolucije bez ponovnog startovanja *Windows-a*.



Proizvođači prateće opreme će imati problema – optimizacija diska, zajedno se atraktivnom grafikom, data je kao standard, mada većina opcija još nije ugrađena.



Ako ne možete da preželite *Windows 3.1*, tu se *Program Manager* i *File Manager*. Možete čak potpuno ukloniti *File Cabinet* i instalirati *Program Manager* kao „shell"

se pojavljuju na desktopu kao i lokalni i možete kreirati veze (*links*) sa njima. Novost je i komunikacija putem modema. To nudi i *Windows for Workgroups*, ali samo kao klijent NT sistema. *Chicago* će moći da funkcioniše kao host, tako da će udaljeni računar moći da pristupi njegovim diskovima kao da su na istoj mreži.

Ostale *Windows* sistemske komande su prilagođene novom interfejsu. Nije potrebno da pomoću programa PIFEDIT menjate *Program Information File* neke DOS aplikacije – dovoljno je da na desktopu otvorite „beležnicu" sa popisom njenih svojstava i preinačite opcije koja se tiču memorije, displeja, taskinga. Boja i karakteristike samog desktopa se menjaju tako što desnim tasterom klikneta na desktop, čime otvarate svesku sa njegovim svojstvima i menjate ih po svojoj želji. Na ražem nivou, postoji nova usavršena konfiguraciona funkcija u folderu CONTROL.SYS, koja daje detaljan grafički prikaz konfiguracije hardverskih drajvera, tako da možete ubaciti *plug-and-play* drajvere uređaja, dakle bez isključivanja sistema.

BOLJI... ALI OD KOGA

Nema sumnje da će *Chicago* biti bolji *Windows* od *Windows-a*, a vrlo verovatno i bolji NT od NT-a. Novi desktop sistem je izvanredan – pravo remekdelo najmodernijeg dizajna udruženog se podrškom se postojeće metode rada i strukture podataka. Sistem se lakše konfiguriše i kontroliše, u velikoj meri zahvaljujući većoj konsistentnosti interfejsa zasnovanog na folderima. Po svemu sudeći postignuta je stopostotna kompatibilnost, a izgleda da su i performanse, koje je inače teško meriti u ranim beta-verzijama, doživele značajan napredak.

Međutim, veliko je pitanje koliko je ovo izdanje po svojoj arhitekturi blisko proizvodu koji će Microsoft konačno ponuditi? Čini se da jeste vrlo blisko. I zaista bi bilo nezamislivo da opcija kao što su **File Cabinet, Links, Properties, Notebook** ili **Recycling Bin** – koje su očigledno u poodmakloj fazi razvoja – otpadnu u konačnoj verziji. Naravno lepo bi bilo videti pravo *preemptive multitasking* jezgro i nešto napredniji sistem oteka, ali je gotovo izvesno da će se i oni, ukoliko se pojave učitavati u memoriju pomoću kopije DOS-a – baš kao i *Novel Netware 4.0*, najnoviji i najnapredniji mrežni sistem za PC računare.

Ukoliko se ova predviđanja pokažu tačnim, onda će *Chicago* predstavljati evoluciju a ne revoluciju, i to mnogo više nego što se moglo naslutiti na osnovu glasina (čiji je izvor recimo i to, uglavnom bio sam Microsoft). Takav sistem će biti lišen glamura koji bi pratio pravi, samosvojni *NT Lite*, ali će on nedostatak sjaja nadoknaditi nudeći kompatibilnost i tak prelazak se sadašnje korisnike *Windows-a*. Međutim, kada spoljašnost (desktop) bude potpuno dovršena, Microsoft se može do mile volje baviti unutrašnjim radovima, što sistemu *Chicago* verovatno i predstoji tokom narednih dvanaest meseci.

NOVOGODIŠNJI PAKETIĆI

ČOKOLADE I BOMBONE SU TAKO DEMODE...
I DEDA MRAZ JE TAKOĐE DEMODE, ZAR NE?
OTKRIJTE SADRŽAJ NAŠIH NOVOGODIŠNJIH PAKETIĆA

29 NOVEMBRA 43/I
11000 BEOGRAD
YUGOSLAVIA

FON: ++38 11 322 19 59
FON: ++38 11 322 35 39
FON: ++38 11 322 53 93
FAX: ++38 11 322 41 20
TELEX 12463 INSOFT YU
PAGING 2222-302/3127

PERFEKCIJOM

art studio **CRNO NA BELO**

Procesor

BEOGRAD, Hadži Nikole Živkovića 2
tel/fax 011/180 739
fax. 632 882
tel 634 548, 632 162
tel 3281 321 (404 405)

DOGRADITE SVOJ DOS SVET

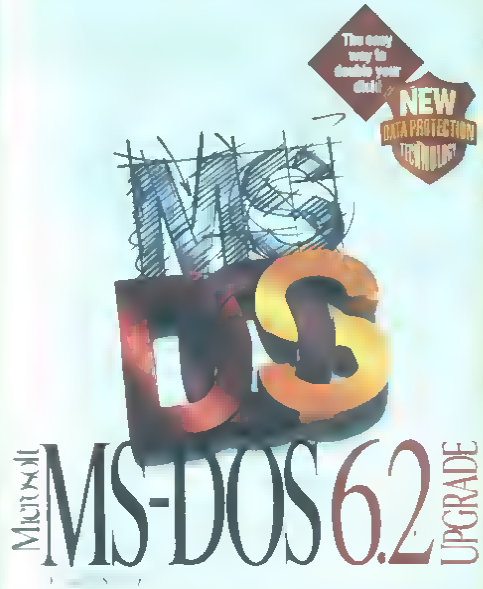
Nakon velikog uspeha DOS-a 6.0 i nekoliko miliona prodatih kopija, Microsoft je u oktobru ponudio dograđenu verziju, DOS 6.2, čiji se noviteti prevashodno odnose na obezbeđenje sistema i algurnost podataka, ali donose i neka veoma zanimljive „sitnice".

Dejan Ristanović

Prošlo je više od osam meseci otkako je Microsoft, na skupu PC User grupe u Moscony centru (San Francisko), na spektakularan način predstavio verziju 6.0 MS DOS-a, najpopularnijeg i najboljeg prodavanog operativnog sistema svih vremena. Promocija koju je, zahvaljujući satelitskim prenosima, gledao čitav svet, bila je tek početak ogromne kampanje nazvane *Upgrade Your World* u koju je Microsoft uložio preko 10 miliona dolara. I, rezultati nisu izostali – prodato je nekoliko miliona primeraka paketa DOS *6.0 Upgrade*, neki kažu mnogo više nego što su i najveći optimisti u Microsoft-u prognozirali. S obzirom da je glavni novitet, program DBLSPACE koji „udvostručava" kapacitet hard diska, ugrađen u same „temelje" DOS-a 6.0, prijavljenih problema je bilo izuzetno malo. Pa ipak, Microsoft je brzo shvatio da se neke „sitnice" moraju promeniti i tako je, oktobra 1993. godine, nastao MS DOS 6.2.

NA VIŠE NAČINA

Do MS DOS-a 6.2 možete da dođete na više načina koji će vas, zavisno od postojeće softverske konfiguracije vašeg računara, različito koštati. Najjeftinije prokšte ako već imate DOS 6.0 – *upgrade* je u tom slučaju besplatan, ali morate da platite desetak dolara za diskete na kojima se isporučuje. Vlasnici modema mogu i taj novac da uštede, tj. da ga „menjaju" za vreme potrebno za prenos datoteke STEPUP se obližnjeg BBS-a (vreme nije baš zanemarljivo, jer datoteka ima oko 1.3 megabajta – na Sezamu smo, zbog vlasnika 1200 bps modema, morali da je podelimo na sedam delova – najlaze se u direktorijumu R:\IBMPC\SYSTEM i zovu se MSDOS62.A01 – MSDOS62.A07). Vlasnici neke od prethodnih verzija MS DOS-a (najpre je računato počevši od 2.11, a onda se Microsoft dosetio i onih koji još imaju „antičku" verziju 1.x, pa je i njih obuhvatio ponudom) za 80 dolara dobijaju *DOS 6.2 Upgrade* i specijalno izdanje knjige neobičnog imena, *DOS For Dummies* (najdobronamerniji prevod bio bi „DOS za neznalice". Inače, u računu sinonima prvi ekvivalent za *dummy* je *idiot*) Dana Gukina (*Dan Gookin*). Za sada se MS DOS 6.2 na prodaje kao poseban paket namenjen onima koji su kupili računar bez operativnog sistema – takvih, bar u Americi, gotovo i da nema. Očekuje se da će, kao i kod prethodnih verzija, poznatiji proizvođači hardvera (Compaq, Dell, Olivetti...) licencirati tu verziju i prodavati „svoj" DOS 6.2. U igri je, naravno, i IBM koji, po ugovoru iz 1980. godine, ima prava na MS DOS i prodaje ga pod imenom IBM DOS – trenutno je aktuelna verzija 6.1 (zapravo nešto prepravljen MS DOS 6.0), zbog koje je Microsoft morao da „skoči" za 6.0 na 6.2.



Instalacija novog DOS-a na računar prošli prilično bezbolno – dovoljno je umetnuti prvu (od tri) 1.44 M disketu, otkucati A:SETUP ili B:SETUP, odgovoriti na par kontrolnih pitanja o konfiguraciji sistema i sačekati da se datoteke prekopiraju. U pripravnosti treba držati jednu praznu 1.2 ili 1.44 M disketu na koju će, u početnoj fazi instalacije, biti prekopirane ključne datoteke iz prethodne verzije DOS-a, kako bi se sistem, u slučaju nevolje, mogao vratiti u prethodno stanje.

Za one koji su se opredelili za „modernaki *upgrade*" stvar je još jednostavnija – komanda SETUP će kreirati potreban direktorijum na C disku, raspakovati datoteke, „zakrpiti" DOS, resetovati računar – dok vi samo sedite i posmatrate. Osim ako...

AKO IMATE BETA...

Osim ako ste DOS 6.0 nabavili od pirata. Kod nas je, na žalost, izuzetno popularna verzija DOS-a 6.0 čiji fajlovi nose datum 22. februar 1993. godine. Ta verzija na prvi pogled mnogo liči na konačnu (VER /R ne ispisuje ništa što bi ličilo na 'beta'), ali to nije komercijalna verzija MS DOS-a. Radi se zapravo o programu koji je Microsoft poslao svim svojim beta testerima na završnu proveru, a istovremeno i raznim kompjuterskim časopisima širom sveta, kako bi bilo vremena da se prikazi ovog paketa napišu i objave u trenutku kada on izađe na tržište. Takve *pre-release* verzije su sasvim uobičajene za softversku industriju i se njima obično nema većih problema, pošto je testiranje uglavnom završeno mnogo ranije; pa ipak, ne preporučuje se njihovo duže korišćenje, ponajpre zato što ih proizvođač zvanično ne podržava. Tipičan primer je DOS 6.0 od 22. februara – program STEPUP iz *Upgrade* paketa će se „začuditi" kako to da, kao jedan od Microsoft-ovih beta testera, niste dobili konačnu verziju DOS-a. Prava verzija, naime, nosi datum 10. mart 1993. godine.

Treba reći da vam je tim odbijanjem STEPUP zapravo učinio lepu uslugu kojo kao takvu treba da cenite – nemojte pokušavati da budete „pametniji" od Bila Gejtsa jer ćete najverovatnije sebi napraviti štetu. Ukoliko se, na primer, „dosetite" da se drugog računara preneseta DBLSPACE.BIN iz nove verzije, STEPUP će uspeti, ali će DBLSPACE biti neispravan, što će najverovatnija dovesti do gubitka svih podataka sa komprimovanih particija. Nećete moći čak ni da vratite stari DOS jer će UNDO prijavljivati da ne može da prepozna instaliranu verziju. Na kraju ćete morati da uradite DEL DBLSPACE.000 – račun za podatke lepostavite piratu koji vam je prodao lošu verziju DOS-a!

Šta, dakle, preporučiti nekome ko koristi DOS 6.0 od 22. februara i njegov DBLSPACE? Jedini razuman savet je detaljan (i to verifikovan!) *backup* čitavog diska i kupovina paketa DOS *6.2 Upgrade*. Onda treba obrisati komprimovana particija, reglamo instalirati DOS 6.2, „dupliranl" disk njegovim DBLSPACE-om i vratiti podatke sa diskete ili trake. Sve u svemu, prilčan posao. Vredi li truda?

Nastavlja se na strani 57

U poslu. Na svakom mestu!

QUADERNO 33
Najmanji Windows PC
na svetu - 386 SXLV

Idealni poslovni pomoćnik:

Pamti svaku Vašu reč i ponavlja sve što ste rekli.
Povezuje Vas sa svetom preko faksa i modema.
Radi za Vas šest sati bez odmora i gubitka koncentracije.
Svežinu i snagu povrati za samo jedan i po sat.
Možete ga puniti i u Vašem automobilu.
U džepu uvek uz Vas kad je najpotrebnije.

Uspešni ga imaju, a Vi ?

olivetti

OLIVETTI ENERGODATA
Palmira Toljatija 5, Novi Beograd, Tel (011) 698.512, 692 007 Fax (011) 695 912

CIM

Koristite moć savremenog znanja !!

CAL

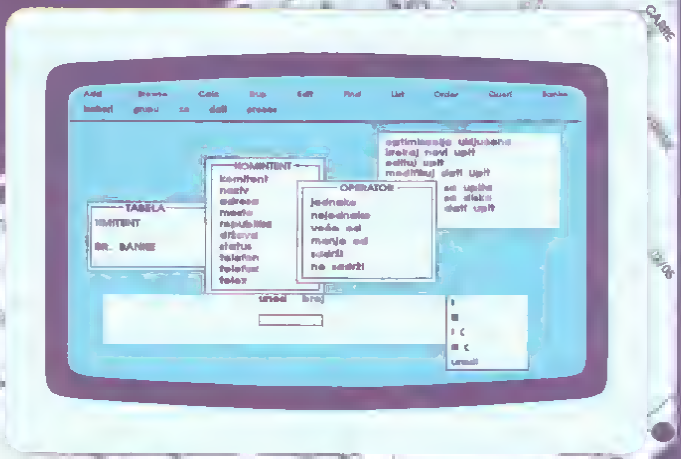
CAM

CAD

CAL - Computer Automated Logistic, Kompjuterska automatizovana podrška proizvodnji

Udahnite dušu Vašem CIM informacionom sistemu, koristite KUP - M.

KUP - M je programski paket koji čini familiju kompjuterskih aplikacija povezanih u jedinstvenu celinu koja korisniku omogućava donoše nje optimalnih odluka o aktivnostima koje se svakodnevno odvi jaju u proizvodnim preduzećima . Tako se u predu zeću stvara okruženje koje svakom pojedincu omogu ćava da maksimalno doprinosi zajedničkim ciljevima preduzeća , Posebna pažnja je posvećena modu lima koje doprinose korišćenju standarda pri aktivnostima koje se odvijaju u predu zeću.KUP - M je CAL moduli u okviru CIM iš - a. KUP - M je izveden tako da se uključuje u proizvodnju i postoje nje integralni deo



SOFTVER

Ciljevi upravljanja proizvodnjom
Glavni produkcioni plan MPP
Glavni produkcioni raspored MPS
Nabavka
Programiranje
Proizvodnja po narudžbini
Predskazivanje potrošnje
Planiranje resursa MRP
- tehnologija
- inženjering
- opterećenje
- rute
- kompletan izgled
- grupna tehnologija
Tehnologija
Radni centri
Održavanje, efektivnost
Inventar (DNV)
Prikupljanje podataka informacija
Upravljanje kvalitetom QCCM
Merenja
-ISO 9000
Kontrola
produkcionih aktivnosti PAC
- otvoreni radni nalozi
- praćenje performansi
Tretmanski centar
- cena proizvoda
- glavna knjiga
- knjigovodstvo dobavljača
- knjigovodstvo kupaca
- plata
- upravljanje kretanjem finansija
- upravljanje ulaganjem
Ostalni programi
- Report Writer

VEZA SA DRUGIM SOFTVEROM

WORD
EXCEL
AUTO - CAD
ORACLE
WINDAX PRO,...

VEZA SA HARDVEROM

Touch terminal
Bar kod reader
Merni instrumenti

I
JOŠ MNOGO TOGA,
I
SVE TO POVEZANO
U
JEDINSTVENU CELINU

JITEX
Obilakovska 57/a,
Beograd
tel/fax: 647-141,
tel: 650-777

NIJE LAKO S BROJEVIMA

Borland još uvek na posustaje u grčevitoj borbi sa Lotus-om i Microsoft-om za primat na tržištu radnih tabela. Najnovije verzije Quattro Pro za Windows, sa oznakom 5.0, je dovoljno kompleksan, obiman i moćan program za sve vrste poslova u ovoj oblasti – od obrade i prezentacije podataka do najauptinijih analiza.

Vladimir Stamenović

Kad god stigne nova verzija *QuattroPro*-a, moj urednik se odmah javlja se veću da ima „vruću" stvar i zaista, ovoga puta smo dobili najnovije verzije *QuattroPro*-a za *Windows* i za DOS, i to obe sa oznakom 5.0. S obzirom da ove godine nismo bili u mogućnosti da detaljnije pratimo šta se dešava na tržištu radnih tabela, bili smo prilično zbunjeni kad smo videli da je *Quattro Pro for Windows* (*QPW*) gotovo preko noći sa verzije 1.0, o kojoj smo pisali u par navrata, došao do verzije – 5.0.

Izgleda da se vraćaju vremena kada smo prvo dugo, sa divljenjem posmatrali kutiju sa novim softverom, pa je tek onda pažljivo otvarali. Program se isporučuje u uobičajeno velikoj kutiji za Borlandove proizvoda, u koju je smešteno dosta dokumentacije i 6 disketa od 3.5", 1 44 Mb.

Od dokumentacije, pakovanja sadrži:

- *User's Guide*, kompletno uputstvo za rad se programom i to na 638 stranica!!!
- *Building Spreadsheet Applications*, potpun pregled makroa, funkcija i ogromna pomoć u gradnji sopstvenog korisničkog interfejsa. Knjiga ima 583 stranice.
- *Database Desktop Guide*, uputstvo za rad se bazom podataka, na 139 stranica.
- *Workgroup Desktop Guide*, uputstvo, na 83 stranice, za razmenu *QPW* tabela se drugim članovima radna grupe, a sve to preko različitih komunikacionih servisa, čak i lokalnih mreža.
- *Quick Reference Guide*, kratki pregled (više) programa, na 50 stranica.

E, sad je na redu instalacija programa. U uputstvu se navodi da je potreban AT računar sa najmanje 4 Mb RAM-a i 10 Mb prostora na disku, S oko 7 7 Mb bez helpa. Kada sam pisao o *QPW* ver. 1 0, kritikovao sam način i brzinu (35 min) instalacije programa. Prošlo je toliko vremena, a stvari se nisu promenile nabolje. Na 386-ici se 4Mb RAM-a oslobodio sam oko 12 Mb prostora na disku. Instalacioni program je manje-više uobičajen – pokreće se iz *Windows*-a, a prate ga lepe slike. Sve je trajalo oko 70 minuta, da bih na kraju dobio šamar – nema mesta na disku?! Ponavljao sam proces nekoliko puta, dok nisam otkrio da je za instalaciju potrebno nešto preko 20 Mb na disku, a posle program zaista zauzima oko 12 Mb. Poslednja instalacija je trajala tačno 65 minuta. Posle instalacije treba obavezno izaći iz programa, resetovati računar i tek onda nastaviti rad.

Da ne zaboravim, program može da radi i bez miša, ali je to onda robija a ne posao. U ovom programu miš je car.

PET RADNIH PROSTORA

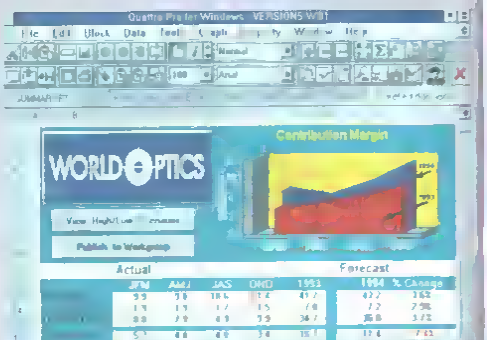
Kada startujete program i kada se tabela, koja se ovda zove *notebook window*, pojavi na ekranu, biće zaista zadivljeni izgledo izgleda fantastično i dosta je unapređen u odnosu na prethodnu verziju(e). U suštini, opšti koncept je stari, ali su neke stvari dodate, a neke poboljšane. Evo šta se vidi na ekranu:

- u prvom redu je pun naziv programa (*QPW*),
- u drugom redu je linija menija (menu bar) sa osnovnim komandama, koje koriste padajuće menije, kao i sve standardne *Windows* aplikacije,
- u trećem redu je već poznati *SpeedBar* koji sadrži dugmiće za aktiviranje najčešće korišćenih komandi,
- novost je u četvrtom redu, takozvani *Productivity Tools*, drugi *SpeedBar* koji sadrži još dugmića za brže izvršavanje komandi,
- ispod njih je *Input Line*, linija se unos i ispravka unetih podataka,
- na samom dnu ekrana je *Status line*, gde se vida informacije o programu (READY, CALC..),
- u desnoj polovini donjeg reda, ispod i sa desne strane tabele, nalaze se skroli barovi koji, kao i kod svih *Windows* programa, omogućuju kretanje po tabeli.
- prostor između ovih, gore navedenih, linija je *notebook window*, odnosno prostor u koji unosite vaše podatke.

Važno je napomenuti da tabela ima 256 „podtabela", odnosno tri dimenzije. One se nazivaju NOTEBK1.WQ2, sledeća je NOTEBK2.WQ2, i tako dalje. Nastavak, ekstenziju, možete sami izabrati. Dopadljivo je relativne promene aktivne tabele. Naime, svaka od njih ima crex kad u rokovniku i obeležena je slovima. Već pretpostavljate da svakoj od 256 tabele možete dati i sopstveno ime, što je u kreiranju aplikacija izuzetno korisno. Na ekranu je vidljivo samo nekoliko oznaka za tabele, pa se strane postoje strelice koja omogućuju prietup i ostalim. Za prelazak iz jedna u drugu tabelu dovoljno je da mišem kliknete na odgovarajuću oznaku i stvar je gotova.

S obzirom da su komande u meniju gotovo uobičajene, ovo samo kratkog pregleda šta se a njima može uraditi:

- **File** komande je prva po redu i služi za bantanje datotekama, štampanje dokumenata, izlaz iz programa, a predviden je i *Windows File Manager*
- **Edit** sadrži osnovne komande za lokaciju i ispravljanja grešaka, omogućuje kopiranje i premeštanje podataka, preko *Windows Clipboard*-a, te vezu sa drugim aplikacijama (DDE, OLE).
- **Block** sadrži komande koje rade direktno sa blokovima, oblastima ćelija. **Move** i **Copy** rade isto kao i DOS tabelama. Deseti klik će omogućiti razna podešavanja sa označeni blok.
- **Data** komande služe za sortiranje ili pretraživanje baze podataka i analize tabele se podacima.
- **Tools** sadrži različite komande za upotrebu makroa, snimanja bloka u datoteku, rad se povezivanjem podataka, analizu podataka i izradu sopstvenih *SpeedBar*-ova i dijaloboksova.



Quattro Pro 5.0 objedinjuje neobičnu lakoću korišćenja i izuzetnu moć, koje će zadovoljiti i najveće zahteve u obradi, analizi i grafičkoj prezentaciji podataka iz radnih tabela

- **Graph** služi za izradu grafikona, pregled, povezivanje u *slide show*, a sve to u sopstvenom prozoru
- **Property** služi za razna podešavanja za svaku vrstu objekta.
- **Window** omogućuje različite poglede na istu tabelu, aranžira i sakriva prozore i prikazuje listu otvorenih datoteka.
- **Help** je odlično urađen i zaista pruža veliku pomoć. Informacije su kratke i precizne, jer se odnose na posao koji je u toku.

I jedan i drugi *SpeedBar* predstavljaju, s jedne strane, olakšanje u radu, a sa druge novo opterećenje. Naime, ima ih toliko mnogo da treba dosta vremena da se nauči šta koje dugme radi. Nisu u pitanju samo *SpeedBar* i *Productivity Tools SpeedBar*, već i neki drugi, koji se javljaju u zavisnosti od toga šta radite. Na primer, grafički prozor ima svoj *SpeedBar*.

Pomoću dugmića možete brže kopirati, seći i ubacivati podatke, pokretati svoje makroe, kreirati grafikone; poravnavati tekst na sve tri načina, centrirati tekst na samo u jednoj već i u bloku ćelija, u jednom redu, prikazivati tekst kao bold ili italik, ubacivati i brisati redove i kolone (intresantno je da na morate obrisati ceo red u tabeli već samo jedan njegov deo), automatski prilagoditi širinu ćelije unetom podatku, sortirati u oba smera, automatski ubacite SUM funkciju, popunjavati (**SpeedFill**) blok ćelija podacima koji zavise od onog u gornjem levom uglu bloka, koristiti predefinisane formate tabela, što će pomoći u brzini izrade i kvalitetu izveštaja, **Expert** dugme pomaže de napravite grafikon i nudi neka dodatna poboljšanja.

Ako vam sve ovo nije dosta, *Productivity Tools SpeedBar* ima još 20 dugmića za najrazličitije namene, a daleko bi nas odvelo da i njih nabrajamo.

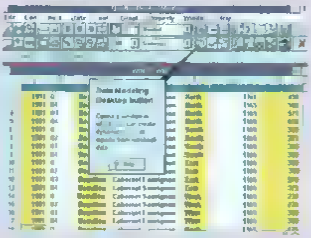
Rad u programu se obavlja u pet radnih prostora. U *Notebook Window*, koji smo ranije nazivali tabelom, unosimo i prikazujemo podatke. Sadrži sve standardne mogućnosti *spreadsheet* programa, uz dosta proširenja i dodataka. Tu se obavlja najveći deo posla – unos podataka, njihova obrada, analize, i slično. Kreiranje, dotenvanje i ispravke grafikona baziranih na podacima iz tabele se obavlja u *Graph Window*. Za pravljenje sopstvenih aplikacija koje će raditi u *QPW*-u, koristićete *Dialog Window*. Poslednja strana u programu, sa 256-te je grafička strana, *Grapha Page*, gde se vrši povezivanje grafikona i slika u *slide show*, ili povezivanje sopstvenih dijalog-bokasova u aplikaciju. Poslednji, peti radni prostor je *Print Preview Screen*, gde se može pregledati red pre štampanja.

SVEŽE NOVOSTI

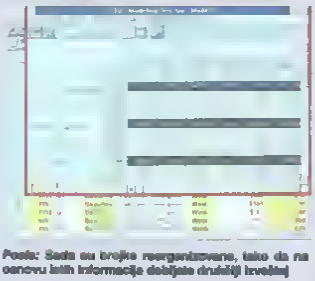
U člancima o prethodnoj verziji *QPW*-a već sam pisao o radu u tabeli, prednostima tradicionalne u odnosu na običnu tabelu, izuzetno lepo rešenom kreiranju i dotenvanju grafikona, velikom izboru tipova, alatima se analizu podataka,. Da na bih nešto od toga ponavljao, u daljem tekstu ću dati samo kratak pregled onoga što je stvarno novo u odnosu na prethodnu verziju(e).

SpeedBar o kome sam već govorio, doneo je neka poboljšanja i veći broj dugmića – ikona. *Productivity Tools SpeedBar* je potpuno nova stvar i o njemu sam već dosta rekao. S obzirom

Opcija *Intelligent Graphs* automatski bira grafik koji najviše odgovara vašim podacima



Pre: Obavili ste analizu, ali želite da ono što ste uradili vidite na drukčiji način; *Data Modeling Desktop* će prebaciti brojke sa jednog mesta na drugo.



Posle: Sada su brojke reorganizovane, tako da na osnovu istih informacija dobijate drukčiji izveštaj

da skoro sve što se nalazi na ekranu predstavlja objekte, desnim klikom na mišu dobićete *SpeedMenu*, čiji izgled zavisi od toga šta radite. Najčešće se koristi u radu se blokom ćelija,

Novo je i poboljšanja Help-a. U *SpeedBar*-ovima ima mnoštvo ikona, ali za svaku možete dobiti objašnjenje u dnu ekrana kada je strelica miša na njoj.

Program se najbolje uči kroz rad. Zato je predviđen *Interactive Tutor* sa kojim možete učiti uz sopstvene podatke. *Tutor* ima svoje dugme i njegovim aktiviranjem dobijamo katalog iz kojeg biramo šta hoćemo da radimo. Prilično lepo urađeno i korisno

Za razliku od pomagala za učenje programa, kao što su *Help*, *Object Help*, *Interactive Tutor* i knjiga *User's Guide*, *Expert* pomaže u nekim stvarima na osnovnom nivou, gde čak i nikakvo učenje nije potrebno. Podeljen je u pet delova.

- *Graph Expert* pravi osnovni grafikon na bazi odabranog bloka ćelija.
- *Scenario Expert* čuva scenario koji je urađen na osnovu vašeg modela.
- *Consolidation Expert* kombinuje blokove sa podacima, konstači jednu od devet statističkih funkcija.
- *Performance Expert* procenjuje da li će kompilacija formula ubrzati vaš sistem
- *Analysis Expert* nudi set alata za analizu podataka, kao i odgovarajući Help.

SpeedFill omogućuje da na osnovu unosa u jednu ili više ćelija, možete lako i brzo popuniti blok sadržajem koji sledi iz tog unosa (broj, tekst, kombinacija). Na primer, ako u prvu levu ćeliju bloka (jedan red sa 7 ćelija) upišete Mon, pa kliknete na *SpeedFill* dugme, blok će se popuniti sledećim nazivima dana u nedelji: Tue, Wed .. (na engleskom)

Kopiranje (**Copy**) i premeštanje (**Move**) su česte radnje u tabeli, a sada imate mogućnost izbora kako to da radite. Najčešći, najlakši i najbrži metod je *drag-and-drop*. Možete koristiti i *Windows Clipboard*, koji ima svoje prednosti u tome što možete ubaciti istu informaciju na više mesta, ili kopirati podatke u i iz drugih *Windows* aplikacija. Prednosti **Block** meni komandi su u tome što se blok može premeštati i kopirati po imenu, a ime se može upotrebiti u makrou. Tu je i **Model Copy** opcija, koja omogućuje da se vidi šta se dešava sa drugim ćelijama u bloku, kada blok koji kopirate sadrži apsolutne reference.

Spell Checker je poslednje što bih očekivao u *spreadsheet* programu. Provrće tačnost zapisanih reči u odabranom objektu, bilo bloku ili grafikonu. Zgodna stvar za one koji rade na stranim jezicima, jer izgleda (nije precizno rečeno u dokumentaciji) da se može izabrati jezik sa kojim se radi

Centriranje teksta u ćeliji je stara stvar. Međutim, centriranje teksta u više ćelija u jednom redu je nešto sasvim novo. Prilično korisno, jer nas najzad oslobađa one bede od unošenja spejsova isprad teksta da bi naslov tabele bio na sredini.

S obzirom da se *QPW* sastoji od više tabela, korisna je nova mogućnost da se određene tabele obeleže određenom bojom. Na primer više tabele koje se odnose na nabavku robe označićemo jednom bojom, a tabele koje se odnose, recimo, na cene, označićemo drugom bojom.

I u prethodnoj verziji *QPW* je prihvatao datoteke iz nekih drugih programa. Sada je ta lista znatno proširena i proces ubrzan.

Kod učitavanja *1-2-3* povezanih datoteka, *QPW* će promeniti originalnu u svoju sintaksu. Ako ste koristili WYSIWYG ili *Impres* dodatke uz *1-2-3*, i njih će *QPW* prihvatiti sa minimalnim ograničenjima.

ANALITIČKA GRAFIKA

Grafički *SpeedBar* se pojavljuje kada otvorite grafički prozor, a sadrži alate za dotenvanje i korekciju grafikona. Tu su uobičajeni **Cut**, **Copy** i **Paste** koji rade preko *Windows Clipboard*-a, **Import**, za brzi prelaz dijalog- boksu za uvoz grafičkih datoteka (formati su BMP, CGM, CLP, .EPS, .CIS, .GIF, .PCX, .PIC i TIF), paleta boja koje se mogu upotrebiti, dugmići za selekciju pojedinih ili svih objekata u aktivnom graf-

konu, crtanje linija i bokseva, te popunjavanje objekata različitim senkama u različitim bojama.

Kad smo već kod grafikona, treba reći da je izbor veći nego ikada, tako da se svaki tip podataka može predstaviti na najbolji način. Ima 10 osnovnih tipova koje *QPW* deli u pet kategorija. 2-D, 3-D, rotirani, kombinovani i tekstualni.

Novost je i analitička grafika koja izračunava nove tačke i prikazuje ih u grafikonu baz promene podataka u tabeli. Na primer, ukoliko vaša tabela sadrži dnevne vrednosti prodaje, analitička grafika će lako prikazati mesečnu srednju vrednost. Ako dnevna prodaja dosta varira, možete prikazati srednje vrednosti tako da ih približite željenim. Takođe, postavljanjem podataka u linijsku ili eksponencijalnu krivu, dobijate generalni trend. Na kraju, možete napraviti tabelu rezultata analiza.

QPW radi kao DDE (*Dynamic Data Exchange*) klijent i server za podatke i grafiku. Možete kopirati te objekte iz *QPW* u *Clipboard*, a onde ih ubacivati u druga aplikacije kao DDE objekte. U tom slučaju, *QPW* je server. Ili, možete da kopirate DDE objekte iz drugih aplikacija i ubacite ih u *QPW*. Tada je *QPW* klijent. *QPW* može, takođe, da radi kao OLE klijent i server sa podatke i grafiku iz tabele.

Ako vaša tabela ima kolonu za svako polje i red za svaki slog, onda možete kreirati „formular" za unošenje i nalaženje podataka, baz programiranja. Sve se to obavlja preko komandi **Choose Data\Form** – označi se blok sa podacima, uključujući nazive polje i sloga. *QPW* će generisati „formular" i prikazati prvi slog. Posao je lagan i zanimljiv. Odlično urađeno!

Data Modeling Desktop (DMD) je novi, analitički alat za kreiranje izveštaja na osnovu podataka iz tabela. DMD izveštaj vam omogućuje brzi pregled podataka iz više različitih perspektiva. Klikom i povlačenjem redova i kolona sa podacima, izveštaj se vrlo lako menja, tako da možete eksperimentisanjem doći do najkvalitetnijeg rešenja, koje će najbolje prikazati vaše podatke.

Scenario Manager (SM) je još jedna novina u programu. Prikazuje promenu podataka i varijabli ubačenih u model i vrednosti rezultata. Na primer, najgori mogući scenario pokazuje šta možete očekivati od poslednjeg željenog seta promenjivih vrednosti, boljі scenario konsti, kao ulaz, vrednosti koje najviše želimo. SM sa može konstti i za praćenje verzija tabela, sa svim, pa i najmanjim promenama u njima. Odgovarajuće komande omogućuju podešavanje i čuvanje SM-a. Korisno ga je upotrebiti za dalje analize kroz *Optimizer* ili *Solver*.

Consolidator je slična mogućnost koju nuda komande **Tools\Combine**. On kombinuje podatke iz više blokova u jedan, koristeći operatore po vašem izboru. Za razliku od **Tools\Combine**, **Consolidator** omogućuje izbor funkcije kao operatora i dozvoljava da sačuvate podešavanja. Najčešće se koristi za permanentne kalkulacije, mada njegova fleksibilnost može biti korisna u analizama podataka.

NAPREDNA ANALIZA

U ovoj verziji *QPW* sadrži gomilu alata za statističku, numeričku i finansijsku analizu podataka, a nazivaju se *Advanced Analysis Tools*. Kada aktivirate komande **Tools\Analysis Tools**, dobićete – a šta bi drugo – novi **Analysis**

Tools SpeedBar. Evo, samo ću nabrojiti šta sve tu ima.

- **Amortization Shedule** je alat koji generiše raspored isplata.
- **Morgage Refinancing** generiše tabelu sa informacijama o refinansiranju otplata.
- **Exponential Smoothing** eksponencijalno poravna seriju vrednosti.
- **Fourier** alat omogućuje brzu Furieovu transformaciju na bloku podataka.
- **Histogram** izračunava verovatnoću i kumulativnu raspodelu za uzorak populacije baziran na serijama ispitivanja.
- **Moving Average** vraća srednju vrednost za seriju vrednosti.
- **Random Number** vraća slučajnu vrednost iz specificirane raspodele.
- **Anova:One-Way** predviđa analizu odstupanja u jednom pravcu.
- **Anova:Two-Way with Replication** predviđa analizu odstupanja u dva pravca, koristeći više od jednog primera za svaku grupu podataka.
- **Anova:Two-Way without Replication** predviđa analizu odstupanja u dva pravca, koristeći jedan primer za svaku grupu podataka.
- **Correlation** alat daje matrični odnos između dva ili više seta podataka.
- **Covariance** daje matrične kovarante između dva ili više setova podataka.
- **Descriptive Statistics** kreira tabelu sa opisnom statistikom koju karakteriše jedan ili više primera.
- **F-Test** testira odstupanja između dva primera.
- **- Rank and Percentile** je alat koji izračunava i u procentima svrstava po rednom broju svaku vrednost u primeru.
- **Advanced Regression** omogućuje višestruku linearnu regresionu analizu.
- **Sampling** je alat koji daje periodične ili slučajne primere uzete iz bloka vrednosti.
- **t-Test** testira odnos dva mala primera.
- **z-Test** testira odnos dva velika primera.

Ako radite sa velikim tabelama koje sadrže puno numeričkih formula, možete kompajlirati formule, što će rezultirati vašom brzinom rekalkulacije. Za maksimalno iskorišćenje, treba vam numerički koprocesor (386 ili 486) i dodatni megabajt RAM-a za kalkulacije. Korišćenjem ove mogućnosti, a u zavisnosti od veličine tabele i raspoložive memorije, rekalkulacija može biti brža od 100 do 200%. Za početak, koristite **Performance Expert**, koji će analizirati vaš hardver, način na koji koristite vašu tabelu i, onda, predložiti izbor. Ukoliko se nakon kompilacije pojavi neka greška u ćeliji, tipa NA ili ERR, problem je moguće rešiti **Audit** sistemom (**Go To i Errors** opcije).

Jedna od zanimljivosti je i da vaš sistem tabela možete zaštiti password-om. Naravno, morate ga dobro zapamtiti da ne biste došli u situaciju da izgubite podatke.

SpeedBar Designer nudi jednostavan način za kreiranje sopstvenog *SpeedBar*-a, koji ćete koristiti za pokretanje komandi ili sopstvenih makroa. Možete birati dugmiće iz palete ili napraviti sopstvene. Svako dugme ima svoj Object Help za lakšu identifikaciju. Talični SpeedBar možete pozvati i ukloniti sa ekrana u bilo kom momentu.

MAKRO KOMANDE

Za kraj sam ostavio par reči o makroima. To je tek prava stvar. Borlandovi eksperti su mislili na svaku sitnicu i napravili pravi programski jezik od makro komandi. Podeljene su po kategorijama.

- makro komande koje zamenjuju tastere
- komande za rad sa ekranom
- interaktivne komande
- komande koje kontrolišu tok programa
- komande za rad sa ćelijama
- komande za pravljenje sopstvenog korisničkog interfejsa
- komande za rad sa datotekama
- DDE komande
- objektne komande
- ostale komande

Nisam tačno brojao, ali mislim da ih ima više od 150!!! U kombinaciji sa oko 240 funkcija, moguće je napraviti fantastične aplikacije.

Opis rada sa makroima traži poseban članak – za sada napominjemo da proceduru možete prikačiti na dugme iz *SpeedBar*-a, tražiti greške pomoću dibegera, ubaciti ih u biblioteku – posebnu datoteku, i još dosta, dosta toga.

Uprkos ograničenom vremenu koje sam imao na raspolaganju, nesumnjivo je da je *Quattro Pro for Windows* odličan program, najbolji koji sam do sada video (u pre svega mislim na *Lotus Improv for Windows*, koji donosi neka revolucionarno nova rešenja u svet radnih tabela; na žalost, nisam video najnovije verzije *Excela*, *Lotusa* i *SuperCalc*-a). Program radi dosta brzo, ne puca, nema čestih i sporih skrolovanje ekrana kao u ranijim verzijama, brzo snima i učitava datoteke sa diska, rad sa mišem je izuzetno kvalitetno rešen.

Sa dobrom hardverskom opremom, korisnici koji imaju velika zahteve u pogledu rada u tabeli, prezentaciji podataka, grafičkoj prezentaciji i, pogotovu, analizi podataka, sigurno će biti više nego zadovoljni mogućnostima koje nudi *QPW*.

Dvadeset i tri „eksperta" rade umesto vas – ne vama je samo da odaberete oblast koja vas zanima

ADACOM
Čika Ljubine 12, BEOGRAD
Tel (011) 629-233, 344-492, 337-367
Fax (011) 629-233, 337-367

• ADACOM - STANDARDNE KONFIGURACIJE

PC 486-50 VLB	4MB RAM, 250 MB HDD, 5 25" i 3 5" FDD, Color	**3,450.-**
PC 486-33 VLB	4MB RAM, 250 MB HDD 5 25" i 3 5" FDD, Color	**3,100.-**
PC 486-33 ISA	4MB RAM, 170 MB HDD, 5.25" FDD, SVGA Mono	**2,350.-**
PC 386-40 DX	4MB RAM, 170 MB HDD, 5 25" FDD, SVGA Mono	**1,750.-**
PC 386-33 SX	2MB RAM, 170 MB HDD, 5 25" FDD, SVGA Mono	**1,450.-**

Doplatu računate kao razliku u ceni izabrane i standardne komponente

• OSTALO

Štampači			
EPSON LX400		9	450 -
EPSON FX1170	A3	9	1,200 -
EPSON LQ100	A4	24	560 -
EPSON LQ570	A4	24	850 -
EPSON LQ1070	A3	24	1,200 -
HP IV L, toner	1MB	300x300	1,950 -
HP IV, toner	2MB	600x600	3 800 -
OLYMPIA AEG, toner	1,5MB	300x300	1,700 -
Besprekidno napajanje			
UPS MIKOM	500VA		470 -
UPS LONG TIME	550VA		520 -
UPS LONG TIME	1,000VA		1,100 -
Fax-Modem			
INFO RUNNER 9624 MNP5 V42bis int.			220 -
DISCOVERY			260 -
Modemi int/ext DISCOVERY			pozovi
Dodaci			
Miš, DEXTRA POINT			40 -
Miš, GENIUS 6000 podloga, adapter			70 -
Džojstik WARRIOR 5			50 -
Zvučna kartica, SOUND WONDER			160 -
Ručni skener, GENIUS 4500A			280 -
Stakleni filter za monitor			40 -
CENTRONiX kabal			15 -
Kabal za napajanje			10 -
Razno			
Koprocesor 80387-40MHz CYRIX			150 -
Strimer 120MB interni COLORADO			450 -
Strimer 240MB interni COLORADO			700 -
NOVELL APP, NE2000, 16 bit			150 -
Potrošna roba			
Diskete 3M, 1 2MB 5 25"			25 -
Diskete 3M, 1 44MB 3 5"			35 -
Diskete NoName, 1 2MB 5 25"			15 -
Diskete NoName, 1 44MB, 3 5"			20 -
Strimer traka, 60MB/120MB			45 -/60 -
Riboni za štampače EPSON, A3, A4			15 -

• KOMPONENTE

				Vaš izbor
Osnovna ploča				
SX 386-33 MHz			220 -	•
DX 386-40 MHz 128c			300 -	•
DX 486-33 MHz ISA 256c			900 -	•
DX 486-33 MHz VLB			950 -	•
DX 486-50 MHz VLB			1,300 -	•
Memorija				
1 MB SIMM			90 -	•
2 MB SIMM			160 -	•
4 MB SIMM			360 -	•
Hard disk				
QUANTUM	85MB	17ms	300 -	•
CONNER, IBM	170MB	17ms	450 -	•
QUANTUM, WD	250MB	11ms	550 -	•
CONNER	340MB	12ms	650 -	•
QUANTUM	525MB	11ms	1 450 -	•
Flopi disk drajv				
5 25" 1 2MB			130 -	•
3 5" 1 44MB			100 -	•
Kontroler				
IDE AT bus 16 bit			45 -	•
IDE cache (4 x HDD + 2 x FDD)			320 -	•
IDE AT bus 32 bit, VLB			100 -	•
Video kartice				
1024x768 512kB, TRIDENT T-9000			90 -	•
1280x1024, 1MB, T-8900/ET4000			160 -	•
1280x1024,1(2)MB,CIRRUS 32bit VLB			230 -	•
Monitor				
14" Mono ROYAL 1024 x 768			270 -	•
14" Color, NI-LR, DATA STAR 0 28 dot			670 -	•
14" Color, NI-LR, PHILIPS, 0 28 dot			820 -	•
Kućište				
DESK TOP, PS 200W			140 -	•
MINI TOWER PS 200W			140 -	•
MIDI TOWER PS 200W			200 -	•
Tastatura				
102 tastera, KB-102			60 -	•
101 taster, YU/ASCII, CHICONY			80 -	•

MOŽE I BOLJE

Neko je već primetio da su 286 računari izumrli, e da su 386 ne umoru. Jedini pravi izbor u današnje brza vramene eu mešine opremijene berem 486SX procesorom. U neše ruke je ovog puta dospele metične ploče se novih hitom ove sezone – AMD 486 procesorom na 40 megeherca – koji po istoj ceni donosi 20% bolje performense od intela.

Bojan Petrović

Industrija je u poslednje vreme jako brza – na tržište se sve češće izbacuju novi procesori, svaki sa svojim jedinstvenim kvalitetima. Doskora je pravi hit među niskobudžetnim procesorima bio Cyrix 486DLC koji je po ceni 386 računara nudio performanse donje 486 klase. Naravno cene jačih procesora su u međuvremenu pale i on je polako počeo da ispada iz igre zajedno sa ostalim procesorima iz 386 klase. Svi proizvođači se danas trude da što bolje zagaze na 486 tržište a tu se, osim Intela, najbolje izgleda snašao AMD. Njegov sveži proizvod 486DX na 40 MHz je nova karta za dobitak i opet za manje para.

Mnogi ljudi koji se bave hardverom primećuju da je Intel počeo da gubi trku sa konkurencijom. Ne samo da su cene Intelovih procesora počela da padaju (486DX/33 sada već kbima kao AMD 486DX/40) nego ni novi procesori koje izbacuje nisu više tako ubedljivo premoćni. Pentiuma nema nigde. AMD Cyrix i još nekoliko proizvođača najavljuju svoje DX2 varijante a 486DX/50 više nije tako usamljen u svojoj klasi (AMD 486DX/40 mu je jako blizu). Među prodavcima se pak zlurado primećuje da je Intel zbog svog monopolizma i zaslužio da blago propadne.

Bilo kako bilo pred nama je sveža matična ploča se već pomenutim AMD procesorom. Red je da joj kao novitetu posvetimo malo pažnje.

SITAN VEZ

Prvo što amo primetili ne matičnoj ploči PX486 P3 je sjajna tehnologija izrade. Veličine je dve trećine „da by AT" standardne ploče, odnosno 220mm x 250mm. Ne samo da je sve savršeno urađeno nego se proizvođač potrudio za razliku od mnogih drugih da ne njoj lepo ispše šta čemu služi, uključujući i sve mogu će rasporeda džampera (*jumper*). Zapravo sva važnija podešavanja se mogu uraditi i bez završnja u uputstvo koje stiže sa pločom. No kada smo već kod uputstva, moramo primetiti da je ono jako šturo. Ne svojih petnaestak stranica donosi samo najosnovnije podatke kao da je nekoga mrzelo da piše. Na sreću ono što je napisano je lepo urađeno i ne osnovu toga se sve može podesiti veoma lako.

Da se vratimo ne ploču. To je univerzalna 486 ploče se opcijama za sve klase 486 procesora (SX DX i DX2) i Pentiuma (autoru se ipak čini da su Intelovci baš ! mogli da ostave naziv 586 ma kakve to zakonske posledice imalo – ipak nekako zvuči poznatije). To znači da sa ovom pločom imate mogućnost jednostavne zamene procesora, ukoliko kupite ploču sa 486SX procesorom jednostavno dokupite 486DX2 i ubodite ga u podnožje. Na žalost, podnožje nije tipa ZIF (Zero Insertion Force), dakle za lako umetanje već najobičnije što znači da je zamena malo problematičnа (u pekstu se novim procesorom će sići i specijalna naprava za „čupanje" čipa iz podnožja ali ja možda ipak bolje to ostaviti nekom ko ima iskustva). Naravno posto se može menjati CPU može se i podešavati frekvencija rada, i to u opsegu od 25 pa sve do 50 MHz prostim podešavanjem odgovarajućih džampera. To istovremeno znači da ste u slučaju zamena procesora spešni brige oko zamene kristala (postoji i verzija ove ploče sa kristalom kada on određuje frekvenciju ali i mali ema sreću da nju ne dobijemo). Čipset je OPTI i to 82C495SLC i 82C206. Ne ploči je i 256 kilobajta sekundarnog keša, i to u *write through* tehnologiji.

Procesor je sasvim uobičajenog izgleda, standardna veličine (logično) se natpisom „Advanced Micro Devices 80486DX 40", i sadrži celih osam kilobajta jnmarnog (intimog) keša. Prema mnogim tevorima, AMD nije ni pokušavao da samostalno sastavi mikrokod čipa, već ga je preuzeo od Intela tako da se sigurnošću možemo reći da nekompatabilnosti ne softverskom nivou neće biti. Što se tiče performansi (o čemu malo kasnije) veličina internog keša kazuje da je AMD rešio da napravi procesor koji će u potpunosti odgovarati intelovoj liniji uz povećanje performansi na osnovu razlike u kloku. Zapravo postavlja se pitanje zbog čega AMD nije odmah skočio na celih 50MHz? Možda je htao da razlikom u cenu popuni rupu koja postoji ne tržištu zbog nepostojanja 486 procesora na 40MHz i istovremeno pruvče sebi one kupce koji bi inače uzeli ploču se 486DX/33? Jednostavno teško je poverovati da AMD ne poseduje tehnologiju za 50MHz.

Ploča prima do 32MB rasporedenih u dve memorijske danke. Kao što je uobičajeno u ova vremena reč je o SIMM memorijama, veličina 256KB, 1MB ili 4MB. Na ploči mora postojati bar jedan megabajt memonje što znači da morate imati barem 4 x 256KB da biste popunili prvu banku. Ukoliko imate SIMM meorije različih brzina, proizvođač kaže da sponje smestite u prvu banku (BANK0), kako bi sistem pravilno radio.

Verovatno najbitnija osobina ploče je postojanje lokalne magistrale u verziji VESA standarda (VESA Local Bus). Postoje čak tri ovakve utičnice od kojih ja jedna *Master*, jedne *Slave* a jedna se može konfigurisati ili kao *Master* ili kao *Slave*. Ova osobina je veoma retka među VLB pločama i sutoru se čini jako pogodnom. U prvim verzijama ploče bilo je nekih zamerki u slaganju VLB se nekim stanjim BIOS-ima, ali smo mi nabavili novu generaciju koje je bez takvih problema. Osim tri VLB ploča ima i tri standardne 16-bitna ISA konektora i jedan 8-bitni, što je sva zajedno sasvim dovoljno za većinu potreba.

BIOS je, naravno, AMI verzija iz 1992. godine, a revizija iz juna ove godine. Meda je datum ovako svež BIOS ja iznenađujuće skromašan. Od naprednih podešavanja dozvoljave samo podešavanje brzine magistrale (od CLKI/6 do CLKI/3) izbor redoslede ispitivanje diskova pri podizanju sistema i nekaliav hard disk pomoćni program koji za IDE diskove i nema nekog smisla. Naravno da na zaboravimo tu je i zaštita šifrom bilo za svako podizanje sistema, bilo samo pri ulasku u podešavanje CMOS vradnosti. Moramo priznati da za par dana korišćenja nismo uspeli da otkrijemo kako se šifra u potpunosti gasi ali to pripisujemo slaboj dokumentaciji (da li je moguća da ne može?)

Dakle hardverski deo posla možemo zaključiti veoma povoljnom ocenom. Ploča je očigledno zreo proizvod sve komponente su od renomiranih proizvođača tehnologija izrade je perfektna. I uopšte, možemo samo da požalimo i da neka slerieća matična ploča koju dobijemo bude ovog tipa.

ŠTA KAŽE ŠTOPERICA

Prvo što prosečnog kupca interesuje je ponašanje ploče u radu i kompatabilnost. Poiito je reč o novom proizvodu još nedovoljno prihvaćene kompanije u ovoj oblasti rad ja da ploču detaljno istestiramo. Za početak recimo da sma brzinu magistrale podesili na celih CLKI/3, što bi trebalo da de najbolje rezultate. Isključili smo sve *shadow* opcije i ne ploču postavili sve standardna delove koje i naše konstimo. Merenja smo obavili u dve stepena se standardnim disk kontrolerom za IDE disk Conner CP30104 i Cirrus 5422 VGA kartom (što su sasvim standardne komponente srednje klase) i sa Longshibe VLB disk kontrolerom i Cirrus 5420 VLB VGA kartom kako bismo stekli utisak o ponašanju ploče u oda slučaja.

Pre nego što se zabavimo posmatranjem rada periferija, moramo posebnu pažnju obratiti ne sam procesor. Prema frekvenciji bi se očekivalo da se procesorski deo posla ne ovoj mašini u odnosu na standardni 486 na 33MHz obavlja brže za nekih 20%. Ako uzmemo da je AMD uspeo da u potpunosti klonira 486 procesor Intela, mogli bismo veoma precizno predvideti brzina u raznim merenjima. No da ne bismo mnogo teoretisali uzeli smo nekoliko prihvaćenih programa za testiranja procesorska snaga i zapisali dobijene rezultate. Koristili smo *Speed* kompanije „Landmark" PMIPS firme „Chips and Technologes" kao i najcenjenijih PC LABS Benchmark časopisa PC Magazine. Rezultati PMIPS i *Speed* testa su grupisani u tabeli jedan. Odavde se vidi da su naše predviđanje bila veome precizna. *Speed* javlja oko 19% poboljšanja, a PMIPS nekih 21% tako da možemo reći da je AMD 486 na 40MHz u osnovi po procesorskoj snazi je ći od Intel 486 na 33MHz za nekih 20%. U istoj tabeli je data i kolona sa merenjima standardne 486/50 mašine, koja je, opet, za nekih 25% brža od naše ploča. Tu je, radi poređenje se najnižim klasom 486 procesora (ili najvišim klasom 386 tako se kome sviđa), kolone se rezultatima Cyrix 486DLC/33 procesora. Ako sada u

Tabela 1

Procesor	*Speed*	MIPS
Intel 486DX/50	169	16.7
AMD 486DX/40	133	13.5
Intel 486DX/33	112	11 1

Procesorska snaga – AMD je 20% brži od Intela

Tabela 2

Processor	TEST 1		TEST 2	
	Sieve	Index	Sieve	Index
Intel 486DX/50	12.92	3.15	4.36	5.11
Intel 486DX/33	16.40	2.58	5.85	3.92
AMD 486DX/40	20.60	2.34	7.08	3.13

Performanse ISA podsistja

Tabela 3

	256KB/256KB keša		256KB/64K keša	
Video kartа	WTACH	WBENCH	WTACH	WBENCH
Cirrus 5422 (ISA)	16.39	736	12.86	2,711,196
Cirrus 5428 (VLB)	28.29	10,346,384	28.31	8,136,831

U *Windows* okruženju

Ovu tabelu unesemo i cena... videćemo da AMD nudi daleko najbolji odnos cena/kvalitet. AMD 486/40 i Intel 486/33 su otprilike iste cene a Intel 486/50 je za više od 200 DEM skuplji (za više od 30% veća cena 25% bolje performanse). No prosto merenje procesorske snage ne govori sve o mašini. Da bismo ipak bili sigurni u ono što smo rekli uzeli smo i Linpack testove koji su dali veoma indikativne rezultate. U tabeli dva smo zato grupisali i nekoliko uzoraka testiranja raznih ploca koje smo imali prilike da testiramo pa možete videti i sami (testovi 558 jednačina, band 100 sa i bez koprocesora)

Sledeći zadatak koga smo se prihvatili je poređenje performansi periferija. Za početak merili smo poređenja ISA sistema, u nekoliko varijanti. Prvi slučaj je bio ponašanje mašine bez ikakvog cache operativnog sistema instaliranog, bez shadow RAM memorije i sa magistralom podešenom na maksimum. Potom smo dodali shadow RAM za video sistem i posmatrali koliko se ubrzao video podsistem. Istovremeno ovim smo skrpili ove mogućnosti za legalno ubrzanje video karte na ISA sabirnici i time stekli utisak o njenom ponašanju. Potom smo na mašinu instalirali sav softver koji obično koristimo među kojima su QEMM, Norton, softverski keš za disk kome smo dodelili jedan megabajt memorije i celu balansu raznih rezidentnih programa. Opet smo izvršili merenja i konačno smo sve zajedno klasifikovali u grupu zaključka.

Pre svega, rad sa maksimalno ubrzanom ISA magistralom se pokazao vrlo bezbednim i brzim. Cirrus 5422 kartica je na svim testovima pokazala nezanemarljivo ubrzanje u odnosu na ploču sa i386DX procesorom na 33MHz (*Speed* kaže 6301 nasuprot 3561). Disk sistem je radio očekivano (sporo) sa dodavanjem softverskog keša sve se složilo i veome silkasnu celinu. Sa druge strane Cirrus 5428 je u *Speed* testu pokazala čak 11988 karaktera u milisekundi što je zaista divan rezultat. Kontroler za disk sa lokalnom sabirnicom je doneo svojih očekivanih 10 15% ubrzanja što postaje mnogo više u disk intenzivnim aplikacijama. Ovde moramo reći i da smo reda radi testirali i jedan nepoznati VLB disk kontroler koji je davao mnoga lolipop i nestabilnije rezultate. Ne možemo tvrditi da je reč o nestaganju kontrolera i ploče već ni se pre reklo da je u pitanju loš proizvod (mada je tu bio očekivani Promise natpis)

Prelazak u *Windows* je definitivno pokazao prednosti postojanje lokalne sabirnice koja je delje radila fantastično. Merenja su grupisana u tabeli tri i predstavljaju sasvim dobar dokaz da je lokalna sabirnica pravo rešenje za ubrzanje video sistema do zavidnih granica. Sistem je radio lako i brzo bez ikakvih problema ili zastoja. Ako je ranije bilo govora o nestaganju ploča sa ovim procesorom i lokalne magistrale sada možemo slobodno reći da nestaganje nije bilo naproč

KUPITI ILI NE?

Ove dilema je jedna od najtežih. AMD se definitivno potrudio da napravi zaista kvalitetan proizvod, koji po čemu ne zaostaje u odnosu na Intelove procesore. Kao interesantnu činjenicu recimo da Intel 486DX/33 procesor sada košta otprilike kao AMD 486DX/40 ali da ovaj drugi kao što smo već videli nudi 20% bolje performanse. Dakle po našem mišljenju ostaje samo dilema da li uzeti već uveliko proveren proizvod sa slabijim performansama, ili ići u korak sa napretkom industrije i oprobati novog takmaca. Mi se opredeljujemo za AMD procesor ne samo zbog pomenutih razlika već i zato što nismo naši ni jedan jedini razlog za nepoverenje u kvalitet ovog proizvoda.

na SEZAM-u BOJAN

NA PRVI POGLED

Procesori

Dogradnja procesora: ubodi i vozi

Za svakog ko ima računar sa 386Dx čipom, Cyrix-ov *upgrade* procesor aa čini kao aan. Bez njega, jedini način da se preda na bržu mašinu ja zamena čitave matična ploče, III čak celog računara. Ovako, aa procesorom Cx486DRX^2, dovoljno ja aamo izvaditi postojeći i386Dx i na njegovo mesto staviti novi čip.

Cx486DRX^2 je mudro samišljen proizvod. U istovetnom 132-pinskom kućištu u kakvo je smešten Intel 80386DX nalazi se procesor koji razumi 486 set instrukcija, radi na dvostruko većem taktu nego ostatak matične ploče i ima 1 kilobajt *cache* memorija koja, takođe radi dvostruko brža. Procesor 386 konsti set instrukcija koji bismo mogli opisati kao podskup instrukcija i486 procesora, pa tako možemo očekivati da problema sa kompatibilnošću neće biti. Međutim, 386DX nema interni keš i stoga ga ne prihvataju ni matične ploče projektovana za ovaj procesor. Zbog toga su u Cyrix-u morali da razviju deo hardvara ugrađenog u sam čip kao i program koji se pokreće pri inicijalizaciji računara i ima zadatak da aktivira keš memoriju u procesoru.

Procesor za nadgradnju se proizvodi u tri verzija, za matične ploče sa taktom od 16, 20 i 25 MHz. Ovom prilikom je testirana srednja varzija i to u računaru ComupAdd 320. Kao poređenje je poslužio računar sa procesorom Intel 80486SX na 25 MHz istog proizvođača.

Instalacija je prošla brzo. Cyrix uz procesor daje i alatku za samanu (vađenje) starog čipa. Doduše, 386-ica je u test računaru bila čvrsto pribijena u podnožja, pa smo sa morali pripomoći i odvijačem. Stavljanja novog procesora je prošlo bez problema, osim što je trebalo malo više snaga da se savladaju 132 pina.

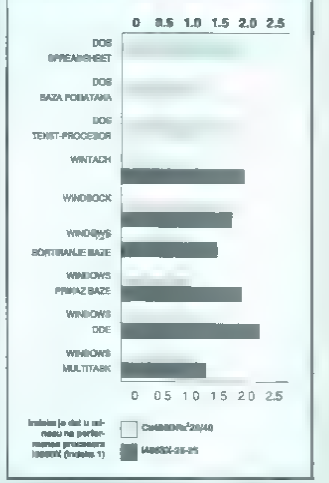
Softver za aktiviranja keš mamnorje može biti instaliran iz DOS-a, *Windows*-a ili OS/2. U vaš AUTOEXEC.BAT dodaće dve linije. Prvi utisak posle startovanja ja da računar radi brže, naročito kod pokretanje *Windows* aplikacija. Ipak, primetili smo i problema, posebno u radu zvučna kartica i, što ja daleko alarmantnije, disk drajva. Gotovo pri svakom zapisu datoteka na flopi i pri jednom pokušaju da mu pristupimo; dolazilo je do problema u radu. Uzročnik ovog problema je pronađan u BIOS-u. Trebalo ja samo isključiti RAM *shadow*.

Većina mašina dozvoljava prepisivanja BIOS koda u RAM, što ubrzava pristup jer je ROM notorno spora memorija. Međutim, stariji BIOS programi često su zavisni od vremena i mogu pokazati slabosti u radu ako ih polerate suviše brzo.

Benčmark testovi pokazuju da ja Cx486DRX^2 poboljšao performanse pod *Windows*-ima do 60 posto, dok je test *spreadsheet* programa pod DOS-om pokazao za 95 posto bolje rezultate. Ipak, sve to nije dovoljno da bi se dostigla i486SX mašina.

Da li Cx486DRX^2 vredi para? Alternativa je nova matična ploča, sa i486SX procesorom na 25 MHz, lokal basom i *OverDrive* podnožjem, čija je vrednost približno ista. Nova matična ploča, najverovatnije, dobro saradivati sa vašim vi

PERFORMANSE UPGRADE



***Winbch* odražava ukupne Windows performanse.**

***Windsock* odražava direktno merenje performansi procesora, displeja, diska i memorije**

deo podsistemom i disk drajvovima, ali ako memorija buda isuviše spora, biće prinuđeni da dokupite i nova memorijska čipova. Time troškovi rastu. Ipak sve u svemu, zamena ploče može vam doneti prednosti moderne tehnologije, ali *upgrade* nikako neće biti tako jednostavan kao sa Cyrix-ovim Cx486DRX^2 procesorom.

Izvor: PC Plus Preveo *Milan Bošić*

HASYS COMMERCE
tel. 021/623-928, 624 501, 616-887
EPSON
LS 400 420 DEM
LX 100 450 DEM
YU SET I CENTRONICS KABL

SAVRŠENA JEDNOSTAVNOST

Visual Basic 3.0 Profsssionsi Edition ss odlikuje bogatstvom ssdržajs, koje gs izdvajs u odnosu ns ostale razvojne siste za Windows. On svojom jedinstvenom koncspcijom dokszujs ds bsjzik ipsk može biti jezik budućnosti i uvodi vss kroz širom otvsra vrats u svst Windows progrsmiranjs.

Zoran Živetić

Pre nekoliko godina, kada se C definitivno učvršćivao kao standardni alat za razvoj program Bil Gejts je tvrdio da je, i porad svega jezik budućnosti bejzik. Doduše, rekao ja da će bejzik preći u objektne jezike, ali će njegova jednostavnost i mogućnost interpretiranja programa koja programeru daje trenutni uvid u rezultat biti osobina koje će ga ponovo učiniti popularnim kao i u samim početcima ličnih računara. Da je neko drugi iznikao ovu prognozu, imao bi samo dve mogućnosti - da ga vreme demantuje ili, što je mnogo manje verovatno potvrdi njegovu vidovitost. Bil Gejts pak ima i treću opciju – da učini da njegova prognoza postaiana stvarnost *Visual Basic 3.0 Professional Edition* zato u poređenju sa ostalim razvojnim alatima za *Windows* deluje kao demonstracija sile – toliko je napakovan osobinama da je teško oteti se utisku da se radi o ličnom hiru Gejtsa – „Obećao sam, i biće!"

OBEĆAO JE I – BIĆE

Moram priznati da je pre instalacije ovog paketa sama ideja da se radi o bejzik programskom jeziku kod mene izazivala odraden otpor. Alati za razvoj *Windows* aplikacija, bez obzira koliko nepradnikbili još uvek na omogućavaju da se ovaj posao obavi potpuno bezbolno pa svaki novi alat pobuđuje pažnju. *Visual Basic* zaista mnogo obećava, ali – zar je moguće da sada treba ponovo da se vraćamo na vremena nedekiarisanih promenljivih biranje imena promenljivih prama tipu podatka koji čuvaju skakutanje po linijama i sličnim tehnikama koje su često bila posledica želje da se ceo interpreter nagura u nazimisliivo mali ROM?

Na sreću, već nakon pregleda nekoliko priloženih primera, polazalo se da je bojazan potpuno neopravdana. *Visual Basic* sa bejzicima kakve poznajemo iz vremena „spektruma" „komodora" III prvih dana PC-ja nema skoro nikakve sličnosti. Doduše tu su i dalje karakteristične imena promenljivih tu je i ono čuveno FOR TO-STEP-NEXT ali ja količina novosti takva da se radi o potpuno novom jeziku. Ipak ono što je glavna karakteristika ovog alata uopšte nema veze sa programskim jezikom – bilo bi u prilično men svejedno da li se koristi C-ovska ili paskal sintaksa, pa otuda i moj utisak da se zapravo radi samo o želji Gejsta da održi obećanja a ne o stvarnoj prednosti ovog programskog jezika. Ipak da krenemo redom

INSTALACIJA

Instalacija *Visual Basic* je potpuno klasična za jednu *Windows* aplikaciju *Visual Basic* se sastoji iz više nezavisnih celina (veliki deo paketa nija radio Microsoft), a korisnik moza da odabere koje celina će instalirati. Ukupan prostor potreban na disku za kompletnu instalaciju je za jeden *Windows* razvojni alat prilično mali – svega oko 30 MB, tako da verovatno nećete imati mnogo dilema koje opcije da izaberete. Pogotovo što svaka od opcija deluje zaista privlačno. Pored osnova, samog *Visual Basic*-a koji zahteva oko 4.5 MB, tu je podrška bazama podataka podrška OLE 2.0 biblioteka ikona programsko „primari" kompajler za help tekstove, dodatna kontrola podrška razvoju sopstvenih kontrola, *Windows API* referentni help, sistem koji objašnjava osnove *Visual Basic*-a, *Crystal reports* sistem za generisnja izveštaja i još nekoliko manjih celina.

Visual Basic se u skladu sa sve popularnijom praksom isporučuje u dve varijanta – standardnoj i profesionalnoj. Nama je na raspolaganju za test bila ova druga. Profesionalna verzija ima podršku za SQL i *Oracle* servera CDK (*Control Development Kit*) *Help* kompajler *Win API help* i nešto više primera i dizajnerskih elemanata za realizaciju korisničkog interfejsa. Nije mi poznata razlika u ceni između ove dve verzije - ako je značajnija, I standardna verzija može biti dobar izbor ukoliko ćete paket da koristi za razvoj aplikacija za „svoje potrebe"

KONCEPCIJA

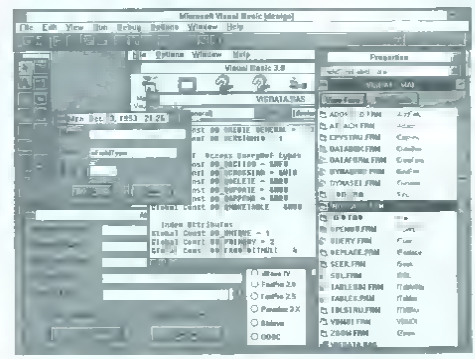
Ono što *Visual Basic* čini jedinstvenim u svetu alata za razvoj *Windows* aplikacija jeste koncepcija. Kada aplikaciju razvijate bez posebnih alata, onda obično počinjete dizajnom osnovnog menija u nekom editoru resursa, zatim sledi kod za inicijalizaciju instance programa, potom konstruisanje glavna petlja za obradu poruka i nakon toga, razrada pojedinih segmenata aplikacija koji sa obično sastoja od forni/kontrola za interakciju sa korisnikom i koda koji obrađuja korisnikov zahtev. Pri tome sa naizmenično krećete između editore resursa kojim sa dirajnirayu pojedini elemenli korisničkog interfejsa i editora u koma pišete kod za obradu.

Kada aplikaciju pišete uz pomoć nekog od *Wizard* alata (poput onoga iz *Visual C++*a) onda počinjsta zadavanjem globalne forma programa (aplikacija u jednom prozoru, MDI i sličnc). uz pomoć resure editora dizajnirate forme za interakciju se korisnikom a potom alat generiše osnovu programa i pomaže vam da pronađete mesto na koje treba upisati kod za obradu pojedinih situacija/poruka. Pri tome ja i dalje pred vama kompletan program u koma možete intervenisali i na, da ih tako nazovemo sistemskim delovima programa, kao što su osnovna petlja za obradu poruke, kod za inicijalizaciju itd

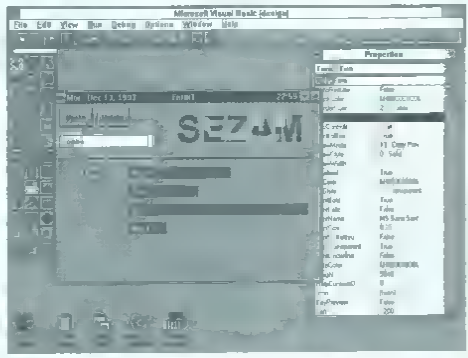
Kada aplikaciju razvijate uz pomoć *Visual Basic*-a, onda je sistamski deo potpuno sakriven – vi se bavite samo kodom koji sa direktno odnosi na samu aplikaciju. Strukture aplikacija je izuzetno čista – svaki program se sastoji iz formi i programa koji obrađuju pojedina poruke unutar forme. Sistem vam nudi potpuno okruženje za dizajn i pisanju i jednog i drugog elementa.

RAZVOJ PROGRAMA

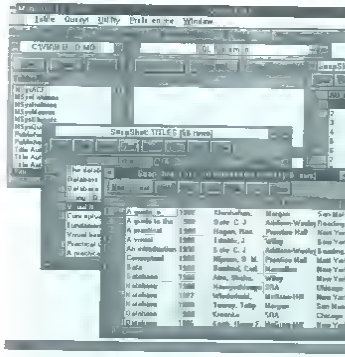
Praktično, razvoj *Visual Basic* aplikacije počinju dizajniranjem forma koja će prva biti prikazana korisniku i koja je na neki način ekvivalent C-ovska main funkcije. Dizajniranje forma u *Visual Basic* u



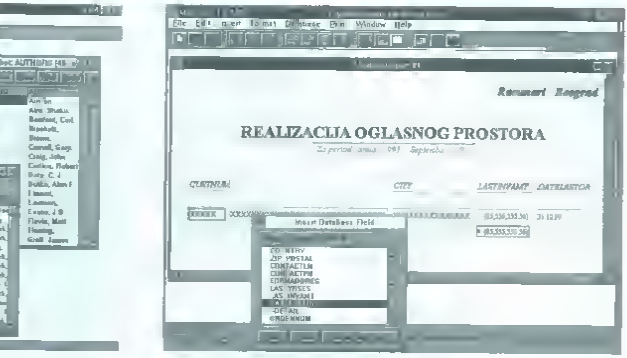
Visual Basic razvojne okolina – puno prozora, malo posla



Forma – osnova atraktivnog korisničkog interfejsa



Sve što je potrebno za rad sa bazama podataka



Atraktivni izveštaji uz *Crystal Raports*

je jednostavnije od rada sa resurs editorima iz drugih razvojnih okruženja. U osnovi, i ovde se radi o prozoru u koji se smeštaju kontrole menija ili dizajnerski grafički element poput linija, površina itd. All ja obezbeđen pregledan način za definisanje osobina svakog elementa.

Konkretno, kada izaberete novu formu iz menija, dobijata prazan prozor. Na ekranu se stalno nalazi i prozor u kome su prikazane osobina (*properties*) ovog objekta u formi liste „naziv – vrednost". Vrednosti kao što su naziv forme tekst u zaglavlju prozora font kojim se ispisuje tekst menjata direktnim izborom polja i upisom vrednosti dok se neka od osobina kao što su veličina prozora, pozicija na ekranu i slično automatski ažuriraju nakon izmena na samoj formi.

Način izmene neke od osobina objekta je izveden na zaista pregledan način. Na vrhu ovog prozora se nalazi trenutno izabrano polje sa listom mogućih vrednosti koja se može otvoriti. Ako na primer izaberete stil ivica za prozor forma, onda dobijete listu od četiri moguće vrednosti iz koje lako birate onu koja vam odgovara. Ako je parametar boja, onda se otvaranjem lista vrednosti otvara paleta boja, iz koje se opet izuzetno jednostavno bira vrednost.

Kada definišete osobine forme onda nastavljate sa raspoređivanjem kontrola ili konstruisanjem menija. Lista kontrola koje *Visual Basic* nudi se kreće od standardnih dugmadi, polja za unos teksta, preko složenijih ali i dalje standardnih varijanti ist polja, pa sve do potpuno orig-inalnih kontrola kao što su mreža 3D kontrola ili izuzetno duhovito realizovane animirane kontrole. Spisak kontrola nije definitivan jer je i osnovna lista sačinjena od ugrađenih standardnih kontrola i dodatnih takozvanih VBX kontrola, koja konstruišu mogu samo da razvijaju U okviru CDK a (*Control Development Kit*) data paleta nalazi se nakoliko primera VBX kontrola koja možete dodati vašoj aplikaciji – automatski će se pojaviti u panelu kontrola i imati svoj spisak osobina.

Elementi grafičkog dizajna se u forma dodaju izuzetno jednostavno. Recimo da želite da formular za unos podataka osvežite logotipom vaše firme. Dovoljno je da dodate **PictureBox** i da među njegovim osobinama promenite **Picture** sekciju i tu upišete naziv datoteke u kojoj je slika. Potom dalje možete doterivati način prikaza ove slika birajući čak i takve detalje kao što je izgled kursora kada se nađe na polju slike.

Sve osobina objekata i mogućnosti koje njihovo podešavanja nude je gotovo nemoguće i pobrojati a kamoli nešto reći o svakoj od njih tpak verujem da će za mnoge biti najinteresantniji **data control** objekat koji omogućava da se iskoristi ugrađena podrška za rad se bazama podatka gotovo bez ikakvog kodiranja. Spisak osobina ove kontrole sadrži i **DatabaseName** i **RecordSource** polja, čima definišete koju bazu želite da otvorite. Sada vam ostaja da kreirate vezu između sadržaja polja i neke druge kontrole – tipično polja za unos teksta. Za svako polje koje želita da prikazata dodajete po jedan **TextBox** i u njegovim osobinama definišete polja **DataSource** i **DataField** čime se sadržaj polja direktno vezuje za sedržaj baze. Veza može biti ostvarena ne samo sa **TextBox** poljem već i sa **Image, Label, Picture, Text** i **CheckBox** kontrolama. Ovo je dovoljno da se dobije aplikacija koja prikazuja sadržaj slogova baza podataka i koja omogućava kretanja napred nazad/prvi-poslednji slog.

Konstruisanja menija je podjednako jednostavno. U posebnom prozoru definiše se opis stavke menija njen naziv i niz drugih parametara – između ostalog i da li želite da se te stavka razvije u spisak trenutno otvorenih MDI prozora. Struktura menija se lako razvija do željene dubina, „navigacijom" kroz strukturu i dodavanjem novih stavki.

Kada završite dizajnerski deo možete odmah pogledati kako interfejs vaše aplikacije izgleda. U panelu sa alatima nalaze se ikone dugmeta čiji smisao asociraju na dugmad na video uređajima – start pauza i stop. Pritisnete start i program počinja sa radom. Naravno dešavaće se samo ono što je interno ugrađeno u sistem a to je reagovanje na kretanja mišem, pritisak na neku od kontrola, unos teksta u neko polje aktiviranja i izbor stavka iz menija i slično. Ono što aplikaciji nedostaje da bi bila kompletna je naravno kod za obradu pojedinih situacija – praktično ono što obično nazivamo programom.

PROGRAM NA SEKCIJE

Visual Basic je organizovan sekcijski. Jednu sekciju čini jedan proces jednog objekta. Na primer, ako vaša forma ima ugrađen meni i ako je naziv jedne stavka menija **OpenFile,** onda jednu sekciju programa čini potprogram **OpenFile_Click** u koji treba uneti kod koji će se izvesti kada korisnik izabera opciju iz menija. Ovo je najjednostavniji slučaj. Kada su u pitanju drugi objekti broj sekcija/potprograma može da bude i veći jer sistem nudi obradu i ostalih poruka poput **DblClick, Get/Lost Focus** i slično. Sistem sam generiše deklaraciju potprograma, što je posebno važno u situacijama kada potprogram prima parametre od sistema. Program naravno ne mora biti vezan za formu već se nit procedura koje su generalno potrebne programu mogu izdvojiti u jedan bezik modul, koji se opet organizuje kao niz potprograma i to potpuno automatski. Uobičajano je i da se da klaracije svih konstanti i globalnih promenljivih i zvada u posebnom modulu/sekciji.

Zahvaljujući potpunoj objektnoj orijenteanosti *Visual Basic*-a kodiranja je prilično jednostavno i programera potpuno oslobađa razmišljanja o internim stvarima *Windows* a. Na primer ako u formi i memo dve **TextBox** kontrola (dve polja za unos teksta) onda fragment koda koji u drugo polja upisuje sadržaj prvog polja izgleda jednostavno

```
polje2 Text = polja1 text
```

Text ja predefinisana osobina **TextBox** kontrola, kao što je recimo i **MultiLine** logička promenljiva čija TRUE vrednost otvara mogućnost unosa višelinijskog teksta.

Ugrađeni objekti su podeljeni u tri kategorije – globalni objekti kao što su **App** (aplikacija) **Screen, Printer** itd zatim interfejs objekti kao što su kontrole forme i MDI forme i **Data Accesa** objekti kao što su **Database Dynaset, Fild, Index, Table** itd. Za svaki objekat definisan je skup osobina, događaja i metoda, što sve zajedno čini pozamašnu količinu informacija kojom je potrebno rukovati da bi se napisao malo složeniji program. Ako se na ovo doda skoro neverovatna količina funkcija koja se kreće u opsegu od uobičajenih do specijalnih finansijskih funkcija, ispada da razvoj programa u *Visual Basic* u ipak zanteva prilično dugo vreme u poznavanja se mogućnostima alata. Ovda je od neprocenjive vrednosti ugrađeni help koji uz pregledsnost već uobičajenu za firmu daje brz pristup traženoj informaciji.

Samo kucanje programa se odvija u režimu trenutne sintaksne kontrole. Čim napustite liniju *Visual Basic* će je analizira i upozorava na greške koje u toj fazi mogu biti uočene. Ako je linija korektna, *Visual Basic* će je i stiskl sredit tako da službene reči budu napisana predefinisenom kombinacijom mala/velika slova i sa predefisanim razmacima i između pojedinih elemenata izraza. Ako ste kao i niju otkucali početak neke funkcija *Visual Basic* automatski otvara novu sekciju vašem početku dodaje odgovarajuću End izjavu i ostavlja vam prostor za unos tela funkcija a sama funkcija se pojavljuje u listi iz koje se kasnije lako može pozvati.

Ovakav način radi bi mogao da deluje odbojno Iskusnijim programerima koji nisu navikli da im okruženje pomaža na ovaj način i da im prikaz na ekranu „skakuće" i menja se u zavisnosti od toga šta otkucaju. Ipak pregledsnost koda koja se postiže na ovaj način je prilična nadoknada za ovaj nedostatak. Kodranje *Visual Basic* aplikacija se retko odvija nezavisno od ostalih elemenata aplikacije

tako će se na ekranu najčešće nalazi mnogo drugih prozora (forms, kontrole i slično) ta editoru ostaje mali prostor u kome je preglednost ključna.

Pri testiranju aplikacije bejzik interpreter dolazi do pravog izražaja. Kada pokrenete aplikaciju u *debug* modu dobijate novi prozor – **Debug**. U ovom režimu program možete izvršavati na uobičajene načine za dibagere – korak po korak se ulaskom (ili bez) u funkciju, pregledati sadržaj promenljivih ili listu istorije poziva do trenutna tačka u programu postavljati prekidne tačka ili zadati stalni prikaz sadržaja promenljivih čak glavna strukcija ovog režima je pomenuti **Debug** prozor u kome možete, poput ostalih implementacija bejzika, interaktivno izvoditi bejzik naredbe. Ako ustanovite da nista korektno dodatih vrednost neku promenljivoj i da će to uticati na dalji tok rada programa, jednostavno vrednost ispravite i nastavite se redom da biste proverili ostatak programa.

BAZE PODATAKA

Već je bilo reči o jednostavnosti s kojom se mogu kreirati *database* aplikacije u *Visual Basic*-u. Ovoj oblasti je na bez razloga, posvećena posebna pažnja u paketu što se može videti već i iz **Currency** tipa promenljive (osmo-bajtna celobrojna vrednost), koja je neophodna u ovakvim aplikacijama, jer omogućuje računanja iznosa bez poznatih problema zaokruživanja kada se koristi aritmetika sa pokretnim zarezom.

Osnovni format baze podataka je *Microsoft Access* ali se database objekat može povezati i sa bazama drugih formata – podržani su *Dbase III* i *IV*, *FoxPro 2.0* i *2.5*, *paradox 3.x*, *btrieve* i generalni metod povezivanja nazvan ODBC (*Open DateBase Connectivity*) koji omogućava da se, ako je obezbeđen odgovarajući drajver, ostvari veza se bilo kojim drugim database serverom. Uz *Visual Basic* se isporučuje podrška za *Microsoft SQL* i *Oracle* servere.

KRISTALNI IZVEŠTAJI

Radna bazama podataka obavezno prati alati za kreiranje izveštaja. Uz *VB 3.0 Professional* izdanje dolazi i *Crystal Reports* (CR) paket firme Crystal Computer Services. Interesantno je da se ovaj paket registruje nezavisno od samog *Visual Basic*-a, ali sa registracijom ništa funkcionalno u paketu na dobija – dobija se podrška od strana proizvođača i ubičajano izveštavanje o novim verzijama.

Crystal Reports omogućava kreiranje dva tipa štampanog otiska – klasičnih izveštaja i labela koja se kod nas obično nazivaju nalepnicama. U obe slučaja, kreiranje počinje izborom baze podataka. Rad se dalje odvija u zavisnosti od tipa izveštaja. Za nalepnice se može definisati potpuna geometrije nalepnice i redosled kojim se više njih štampa na strani (odozgo na dole ili po redovima). Na ekranu se potom dobija prazna forma sa vodećim linijama i labele se polijme baze. Sve što treba učiniti je izabrati polja i potom ga preneti na željeno mesto na izveštaju. Pored polja, na papir se može postaviti linija, ram ili slika (podržani su BMP, GIF, PCX, TIF i TGA formati) što omogućava da se kreira zaista atraktivna forma. *Crystal Reports* nudi i više predefinisanih formi nalepnica – od adresnih preko nalepnica za razna formate disketa, do nalepnica za audio/video kasete.

Za razliku od nalepnica koje se kreiraju krajnje jednostavno, izveštaji zahtevaju nešto više rada ali nude i više mogućnosti. Prazna forma sadrži tri osnovne sekcije – zaglavlja, detalje i dno izveštaja. U svaku od ove tri sekcije se polja postavljaju kao i kod nalepnica jednostavnim „dovlačenjem" iz liste polja baza. Za svako se polje opet može definisati niz atributa. Na primer, može se zadati da se polja na prikazuje ako je jednako prethodnom, može se izabrati tip poravnavanje podatka i do tančina definisati prikaz podatka ukliko se radi o numeriku – od zaokruživanja, preko razdvajanja hiljada, do izbora simbola i mesta pojave valute.

U izveštaj se mogu postaviti i polja koja nastaju izračunavanjem putem definisana formule. U formuli mogu figurisati polja baze kao promenljive obezbeđen je skup velikog broja funkcija (od aritmetičkih preko string i datumskeh do statističkih) i niz operatora, tako da izraz može biti izvenredno složen. Zato je obezbeđene i sintaksna provera izraza da na bi došlo do grešku u trenutku izvršavanja.

Porad toga, polje može de bude i podtotal. I tade se automatsko otvara nova sekcija koja predstavlje „gnezdo" za niži nivo. Konkretno to znači de dobijate mogućnost de dizajnirate ne samo liniju koja sadrži podtotal već i zaglavlje za ovu sekciju.

Porad jednostavnih izveštaja koji sadrže podatke iz jedna baze i koji su relativno nebitni u praksi *Crystal Reports* omogućava lako kreiranje takozvanih linkova – veza sloga tekuća baze sa slogovime drugih baze na osnovu sadržaja ključnog polja. Zahvaljujući ovoj mogućnosti, upotrebne vrednost *Crystal Reports*-a zalazi u oblast profesionalne primene.

I kod izveštaja se, kao i kod nalepnica, mogu dodati i grafički elementi koji upotpunjuju dizajn izveštaja. dok se za sama podatke može birati način prikazivanja koji uključuje na samo font već i ramove i boju.

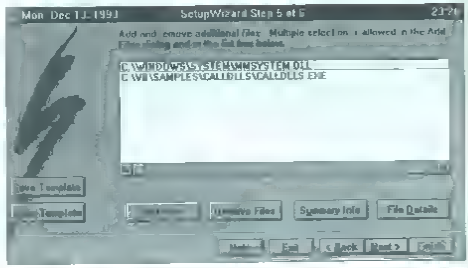
Štampa izveštaja je izvedena u skladu sa fleksibilnošću celog paketa – izveštaj se može „od štampati" u prozor (kada dobijatete prethodni prikaz jednak konečnom prikazu na papiru) u datoteku ili, konačno, na štampač.

GENERISANJE INSTALACIONE DISKETE

„Tačku na i" ovog paketa stavlja *SetupWizard* (SW) – čarobnjak za kreiranje instalacionih disketa se izabranim programom. Kreiranje instalacionih disketa za distribuciju programa bi bez ovog alata bilo skoro nemoguće jer, pored aplikacija na distribucionu disketu treba preneti i niz sistemskih datoteka ukoliko program kodsti neki od dodatnih alata *Visual Basic*-a, kao što su *Cristal Reports* ili red sa bazama podataka. Uz *SetupWizard*, dovoljno je navesti naziv projekta (*makefile* detoteku), koje osobina poseduje paket (prstup bazama, OLE CR itd) na kom i kakvom disku želimo da kreiramo distribucionu verziju paketa, dodatna datoteka koje želimo da uključimo – i *SetupWizard* će generisati sve što treba, konsteći kompresiju datoteka i raspoređujući podatke na više diskata ukoliko na mogu da stanu na jednu.

Sama instalacija je potpuno „a la Microsoft" sa prepoznatljivim plavim prozorom u koma je ispisan naziv aplikacije i prikazan tok procesa. *SetupWizard* instalacioni program generiše u *Visual Basic*-u, pa se njegovom modifikacijom može kreirati drugačji dizajn ili tok instalacija.

Ako niste imali prika da ovaj posao radite ručno verovatno nikad nećete saznati kolika je stvarna vrednost *SetupWizard*-a. Posao koji *SetupWizard* obavlja potpuno automatski inače je strahovito zamoran i traži maksimalnu koncentraciju. Zamislite samo da greškom izostavite iz paketa neku datoteku, to na primetite u fazi testiranja, i zatim krenete sa distribucijom programa. Ovakva greška vas može više nego skupo koštati.



SetupWizard – kreiranje kompletnog setup programa za valju aplikaciju

ALAT BEZ KONKURENCIJE

Pri vrednovanju svakog alata za razvoj *Windows* aplikacija, postavlje se nakoliko pitanja. Prvo je obično pitanje brzine razvoja odnosno produktivnosti programera. Od svega što sam od sada imao prilika da vidim *Visual Basic* je potpuno bez konkurencije na ovom planu – nija mi poznat alat opšte namene kojim bi sa mogla brža i jednostavnije razviti aplikacija. Ipak želim opet da naglasim de je ovde presudna koncepcija paketa a na programski jazik.

Da li Visual Basic omogućuje razvoj bilo kakve *Windows* aplikacije i može il sa ovim alatom napisati složen program poput *Corel Draw-a*? Iako nama teorijskih prepreka da se *Corel Draw* napiše u *Visual Basic*-u, ipak verujem da je odgovor na ovo pitanja – na. Ovda ja sva svega, prepraka sam bejzik kao jezik, koji je u suštini ipak nezgrapan i neproduktivan kada aplikacija zahteva „gomilu kode".

U ovoj poslednjoj konstatacji se krije i odgovor na pitanje kome je zapravo namenjen ovaj peket. Iako će mnogi koji se programiranjem bave iz hobija u ovom paketu pronaci dosta zadovoljstva i po prvi put mogućnost da uđu u svat *Windows* programiranja, rekao bih ipak da su glavni cilj posiovne aplikacije. Uz *Visual Basic* će programer biti u mogućnosh de relativno lako i brzo razvija specifične aplikacije za poslovanja firma koja sa odmiču od monolotna forma 80x25 akrana i konačno počnju da predstavljaju i vizuiano zadovoljstvo za korenika, uz sve prednosti koje nudi uniformen konenički interfejs *Windows*-a.

Na znam da li se iz ovog teksta može naslutiti koliko je bogatstvo sadržaja *Visual Basic* paketa. Dalovi kao što su *Cristal Reports* III *Setup Wizard* predstavljaju već sami za sata izuzetnu vrednost a u kombinaciji se ugrađenom podrškom bazama podataka čine ovaj paket zaista jedinstvenim. Posle iskustva sa *Visual Basic*-om ostaja nam samo da se nadamo da će Bil Gejts davati što više proročanskih izjava!

Korisna adresa

MISOFT
11000 BEOGRAD Skadarska 45
Tel: 011 343-043

PC ⇨⇦ VIDEO/TV NA OSAM NAČINA

ATVista

- Texas Instruments 32-bitni procesor za ploči
- 256-16.7 milione boja i posebno "ALFA" kanal
- 1MB, 2MB ili 4MB video memorija
- Procesorska memorija 2MB do 16MB (za dodatnu ATVMX ploči)
- PAL (CCIR-624) standard, kvalitet za profesionalnu primenu

VIGA

- Potpuna TARGA+ kompatibilnost
- 256-16.7 milione boja
- RGB, kompozitni PAL i S-video ulaz i izlaz
- Specijalni efekti u realnom vremenu (crossfading, linear blending, chroma key, mosaic, logic operation, mirror, wipe, fid)
- Video kvalitet posebno za profesionalnu primenu

VIGA Genlock

- Prebrava VGA video signal u kompozitni standardni PAL video signal
- Mogućnost sinkronizacije za eksternim video signalom (GENLOCK)
- Preklapanje video signala za ulaznim video signalom i Chroma keying
- 2 ulazne video signala
- ET4000 i 1MB DRAM za ploči

VIGA Packer

- JPEG kompresija i dekompresija
- Odnos kompresije do 50:1
- RGB i Windows DLL biblioteke
- 4x8 virtualni drajver za Windows

VIGA VGA

- Prebrava VGA video signal u RGB, kompozitni PAL i S-video signal
- Jednovremeno prikazivanje video i VGA signala
- ET4000 VGA čip za ploči za 1MB DRAM i HiColor RAM DAC
- Posebno filter za stabilan video prikaz

PortaShow

- Eksterni uređaj za konverziju VGA video signala u kompozitni PAL, S-video i RGB video signal
- Jednovremeni prikaz video i VGA
- Posebno filter za stabilan video prikaz
- Rezolucije do 800x600 PAL
- Podržava 16/256/32K/64K/16.7 milione boja
- Dimenzije: 12.5x6.8x2.4 cm

VIGA Window

- 3 kompozitna PAL ili S-video ulaza
- Kompresija i dekompresija "FULL MOTION" video signala u realnom vremenu
- Prikazivanje video signala u prozoru za VGA monitore
- C & T PC video čip za ploči

Software

- 3DSTUDIO 3.0 Kompletan sistem za profesionalnu 3D animaciju
- TOPAS Kompletan sistem za profesionalnu 3D animaciju
- IDO Kompletan program za kretanje 2D slika
- IDO ANIMATOR Profesionalna 2D animacija
- QFX Profesionalni program za rotaciranje i kretanje 2D slika
- PANORAMA 2D animacija i 3D 2D efekata u realnom vremenu

Usluge Elektronske Grafike

-RAČUNARSKA GRAFIKA I ANIMACIJA, RAČUNARSKA PODRŠKA REALIZACIJI TV EMISIJA, RAČUNARSKA PRIPREMA ZA ŠTAMPU, RAZVOJ GRAFIČKOG SOFTWARE-A, ZASTUPSTVO OPREME ZA RAČUNARSKU GRAFIKU, ARHITEKTONSKA VIZUALIZACIJA I PREZENTACIJA, PODRŠKA ZA PREZENTACIJE, MARKETING, OGLAŠAVANJE U MEDIJIMA

-OSTALE USLUGE: Snimanje vašeg materijala na profesionalnim video uređajima, skeniranje kolor fotografija i dijapozitiva, post produkcija, konvertovanje svih video i kompjuterskih formata, praćenje u štampi

Elegra

Narodnog Fronta 45 - Tel/Fax 643-029, 642-055/140

GRAMATIKA NIJE BAUK

Takst-procssori odavno nisu jedina sistka za pisanje - uz njih idu programi za proveru pravopisa, obični rečnici i rečnici sinonima, a našto kasnije su im sa pridružili i mnogo „pametniji" programi za provaru gramatika. Po mnogim ocanama, najbolji među njima ja Grammatik 5.

Ranka Jovanović

Na tržištu softvera ima sve više programa koji se jednom rečju nazivaju „writer's tools", a u te spisateljske alatka spadaju i programi za proveru gramatika i stila tekstova pisanih na engleskom jeziku. Prodaju sa samostalno ili u paketu se nekim od popularnih tekst-procesora, a najpoznatiji su, uz *Grammatik 5*, *Correct Grammar*, *Power Edit*, *RightWriter*, *The Writer's Tool kit* i Wordperfect ja nedavno odlučio da svoj čuveni tekst-procesor učini moćnijom alatkom za pisanja i pridruži mu *Gramatic 5* (ranije vlasništvo kompanije Reference Software koju je Wordperfect kupio). *Grammatik* ja dugo bio bestseler u ovoj oblasti softvera, made je u izvesnim penodima bilo i boljih programa od njega – tako je 1991 *Grammatik IV*, verzija 2.0, bio tek na trećem mestu rang-lista. Međutim, *Grammatik 5* ja dopunjen i unapređen i, prema rezultatima testiranja američkog časopisa *PC Magazine* objavljenim u maju ova godina, neprikosnoveno vlada na polju elektronskih stručnjaka za gramatiku i stil engleskog jezika. *Grammatik 5* se prodaja i samostalno u verzijame za DOS i Windows, a mi smo ga dobili sa najnovijim *Wordperfect*-om 6.0 za DOS (čiji paket opterećuja se skromnih 2,2 MB)

Grammatik 5 sa poziva iz Wordperfect-a izborom opcije **Writing Tools**. Prihvata tekstova iz većine tekst-procesora, u standardnom ASCII formatu. Pre nego što mu prepustite svoje delo, možete donekle uticati na njegova kriterijuma, izborom jednog od ponuđenih stilova (od opšteg stila, poslovnog pisma i izveštaja, preko tehničkog i pravnog, do novinarskog, pa čak i stila umetničke proze). Takođe je moguće odrediti stepen formalnosti (*informal, standard, formal*). Kada sa pokrene interaktivna provera, ekran se deli na dva prozora – u gornjem je tekst koji se čita, a u donjem sa vrlo uočljivo navode greška, pa zatim komentar ili savet.

KAKO SMO TESTIRALI

Da bismo proverili mogućnosti ovog programa, pripremili smo tri sasvim različita vrste engleskog teksta – prvi je bio prevod zvaničnog sudskog dokumenta, drugi kratak odlomak iz proze savremenog američkog pisca (Joseph Heller), a treći tekst sastavljen upravo za ovu priliku, sa namernim greškama u pravopisu (*spelling*), interpunkciji i upotrebi gramatičkih struktura. Trudili smo se da to budu tipične greška koje, prema iskustvu primenjene lingvistike, prave oni kojima engleski nija materinji jezik. Pri tome smo bili svesni da ja samo prvi tekst realan zadatak za elektronskog lektora, ipak, htell smo da ispitamo i kako se nosi sa kreativnom prozom kada već prihvata prozni stil (*fiction*). Sa trećom vrstom teksta smo želeli da provenmo koliko ovaj i slični programi uopšte mogu da posluže kao pomoćno sredstvo u učenju engleskog jezika.

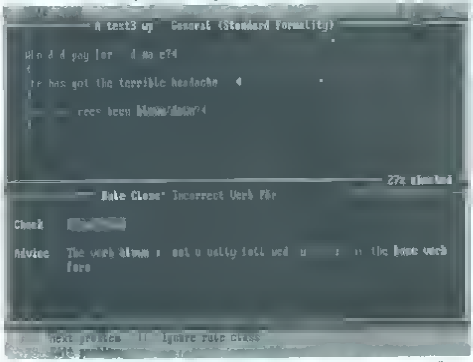
U procesu vrednosti ovakvog programa, važno je obratiti pažnju na sledeće reakcije: broj grešaka koja program pronalazi, broj promašaja, odnosno slučajeva kada program ne uočava grešku, vrstu i kvalitet saveta ili sugestije koji program daje za ispravljanje greške, broj lažnih uzbuna – slučajeva kada program ukazuje na nepostojeću grešku ili preporučuja ispravku koja povlači za sobom novu grešku. Jedan od kriterijume pri vrednovenju ja i odnos pogodaka i promašaja – naravno, što je veći rezultat u korist pogodaka, to je program bolji i pouzdaniji.

Za prednost koju *Grammatik 5* ima u odnosu na ostale „gramatičare" zaslužan ja u velikoj men novi tip mašina za proveru ispravnosti jezičkih struktura po imenu Mor*Proof – sadrži specijalni rečnik od 100,000 reči zasnovan na morfemama (korenska reč ili najmanje jezička jedinica koja nosi značenja). Kada traži neku reč, recimo *bringing* ili *brought*, Mor*Proof prepoznaje njenu vezu sa morfemom *bring*, a zatim pretražuja celu porodicu reči kojoj one pnradaju.

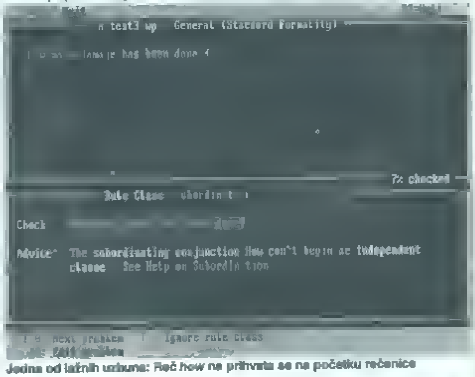
NE BOJTE SE LAPSUSA

Pokazalo se da *Grammatik* znalački ukazuje na osnovne reklt bismo notorne greška u upotrebi gramatičkih struktura. To su izostavljeni ili nelogični glagolski nastavci, neslaganje subjekta i predikata u licu i broju, upotreba nepotpunog glagolskog oblika na mestu predikata, komplikovana glagolska struktura koju uspešno zamenjuja jednostavnija. Dakle, uz *Grammatik 5* nsma bojazni da ćete negde izostaviti ono englesko **a** ili **an** u trećem licu jednine, ili da ćete vam se omaći do umesto **does**, jer će vas program napogrešivo ali veoma taktično upozoriti da subjekat u jednini, logično, zahteva da i predikat buda u jednini. Nakon što vam je jednostavno i sažeto pravilo dato na znanja, nudi sa automatska zamena. Tako u rečenici „He **tell** me that. ", *Grammatik* izdvaja i obeležava crvenom bojom reč **tell**, i daje pravilo odnosno savet: „The singular subject (he, she) takes a singular verb not the plural verb tell." Ako se složite i odaberete zamenu, **tell** će biti zamenjeno se **tells**.

Veliki kamen spoticanja u savladavanju engleskog jezika je upotreba i izostavljanja određenog i neodredenog člana ispred imenice. U



Grammatik na prepoznaje „frazne glagole", koji su veoma karakteristični za engleski jezik



Jedna od lažnih uzbuna: Reč *how* na prihvata se na početku rečenice

tome je *Grammatik* izuzetno pouzdan savetnik – u slučaju da napišete „My best student deserved excellent grade" on će vas upozoriti da nešto nedostaje „an article or other modifier usually precedes the word **grade**". On zna da ispred nebrojive imenice ne može da se upotrebi neodređeni član kao i da posle broja **two** ne može da sledi imenica u jednini **woman**, već samo oblik za množinu **women**. Nija mu promakla ni kombinacija **much coffees** – poučio nas je da se **much** ne konsti uz brojive imenice kao što je u ovom slučaju **coffees**.

Posebno korisna opcija za učenje engleskog jezika, u kojem je ponekad veoma teško odrediti vrstu reči (pomenimo samo reč **right**, koja može biti pridev, imenica, glagol ili prilog) jeste sposobnost *Grammatik*-a da analizira rečenicu i prikaže je u vašem dokumentu sa označenim vrstama reči

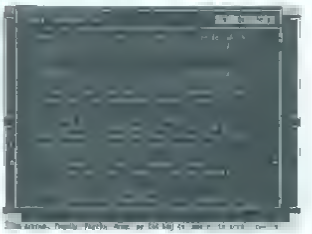
LAŽNE UZBUNE

Ali, ako pustite mašti na volju i upotrebite neku neobičnu kombinaciju reči, *Grammatik* se neće snaći, što i nija iznenađujuće. Ipak, pogledajmo kako reaguja u takvom slučaju. U rečenici „ If I am tired or dejected from a night of lies or booze or just **plain nerves**, I can ", on je pridev **plain** interpretirao kao glagol sa imenicom **nerves** kao objektom, pa je ssavim nesuvislo upozorio: „This verb does not usually take an object."

U slučaju iole originalnije proze biće, na žalost, suviše ovakvih lažnih uzbuna. Vrlo je dosadne napomena „Long sentences can be difficult to read and understand Consider revising, so that no more than one complete thought is expressed in each sentence." No dobro, suviše bi bilo očekivati od *Grammatik*-a da razume spisateljske slobode. Srećom ovo ili bilo koja drugo pravilo možete isključiti tako da ga *Grammatik 5* nadalja ignoriše. Štaviše, postoji mogućnost da kreirate vlastiti stil za pronalaženja grešaka, koji će najbolje odgovarati vašim potrebama. Time *Grammatik* neće postati bolji lektor, ali ćete se barem donekle osloboditi nepotrebnih ili nesuvislih komentara

Ali, budimo realni – kome ja ne ovim našim prostorima engleski jezik potreban da bi pisao kreativnu prozu? Ogromnoj većini ljudi on služi za pisanje poslovnih pisama ili stručnih tekstova iz oblasti koju dobro poznaju. U tome im **Grammatik** može biti od pomoći, što se pokazalo i na našem primeru, kada smo mu prepustili na ispravku prevod jednog običnog sudskog uverenja. U pronalaženju pravopisnih i interpunkcijskih grešaka se pokazao kao pravi ekspert, koji vodi računa i o najsitnijim detaljima – ukazao je na grešku u ispisivanju datuma (write May 8, not May 8th), savetovao da brojeve ispišemo slovima a ne ciframa predložio da proverimo kako se piše **multi-vendor** (a hyphan is not usually used with this prefix) sasvim ispravno upozorio da se „if" rečenica odvaja zarezom. Jednom rečju, ako solidno poznajete engleski jezik i oslonite se na ekspertizu programa *Grammatik 5*, nećete morati da se stidite svoje korespondencija ni tekstova koje prepadaju nekom od standardnih stilova. Osim toga, sa programom koji istovremeno tako pouzdano proverava i pravopis i gramatiku uštedećete dosta vremena.

Za elektronske gramatičare uopšte, pa i za najboljeg među njima, *Grammatik 5*, tvrd orah



Ako vam nije jasno šta je elektronski lektor hteo da kaže, možete otvoriti malu gramatiku i naći detaljnije objašnjenje.

predstavljaju i „domaći zadaci" iz engleskog jezika, u kojim se sreću tipične greške pogrešno postavljano pitanja (Who **did say** that?), pogrešno odabrano glagolsko vreme (**Have** you **had** good whather last weekend?), pogrešan izbor predloga (Put the desk **before** the window). Sposobnost da se uoče ovakve greške bi zahtevala da se sa morfološkog nivoa oda na viši, morfo-sintaksički nivo, što bi bio značajan korak u lingvističkom smislu a verovatno ogroman u softverskom, jer bi podrazumevao posmatranja znatno šireg konteksta, pa tako i daleko veću bazu na koju bi se program oslanjao. Tek tada bismo mogli očekivati da *Grammatik* "zna" da iza glagola **depend** dolazi prepozicija **on** a nikako **of**, kako naši ljudi, prevodeći direktno sa srpskog, obično pišu. Sa svog sadašnjeg stanovišta, *Grammatik*, naravno, ne nalazi ništa loše u strukturi **he was at home**. Kada bi bio u stanju da skenira širi kontekst, lako bi uočio da je to deo fraze „if he was not cross with me" i tada bi sigurno savetovao da se **was** zameni se **were**.

Isti problem ilustruje i rečenica „He **made me to think** that I had hurt his feelings" Glagol **maka** se ponaša prema strukturalnom obrascu koji je poznat svakom malo naprednijem daku – maka + object + bare infinitive – all *Grammatik* preko ove greške prelazi

Doduše, ima i trenutaka kada *Grammatik* zaista briljira i daje veoma pametne sugestije. Za rečenicu „She suggests to go to the party" komentar ja bio „The verb suggest is usually followed by a present participle, not an infinitive" No ako *Grammatik* zna po kojem obrascu se ponaša **suggest**, zašto na poznaje obrasce za **make**? Odgovor ja verovatno to da glagol **make** ima nebrojeno mnogo značenja i ulazi u kombinacije sa raznim strukturama, dok ja glagol **suggest** u tome sasvim isključiv, pa je za njega lako ustanoviti pravilo

Ime i propusta koji bi se relativno lako mogli otkloniti. Na primer, prosto je neverovatno da *Grammatik* na prihvata upitnu reč **how** na početku rečenice tretirajući je isključivo kao zavisni veznik – dakle, ne zna ni za **how meny** ni za **how much**. Takoda ne prepoznaja frazne glagole (sasvim ga je zbunio drugi deo glagola **blow down** – smatreo ja da sa radi o drugom glagolu u infinitivu!), kao ni komparativ prideva (nema komentara za **more warm, more nice**)

IMATE LI STILA?

Što se stila tiče, *Grammatik* savesno vodi računa o tome da ne ponavljate iste reči, da ne počnjete rečenicu uvek na isti način da vam paragrafi ne budu suviše kratki, insistira na jednostavnosti i preciznosti U tome će nekada biti konstruktivan – na primer kaoe predloži da obik **would have been able to help** zamenimo se jednostavnijim **would have helped** a nekada sitničav kao u slučaju reči **impelementation** koju savetuja da izbegavamo

Stil govornog jezika se mnogo razlikuja od pisanog stila, a pogotovo od zvaničnog, službenog ili poslovnog Upravo to su jezička finese koja je najteža razlučiti i dobro je što se u tome odnekla možete osloniti na *Grammatik* - on će vas podući da je u odredenom kontekstu prihvatljivija relativna zamenica **which** nego **that**, da izostavite reč **rather**, koja je od preterana upotrebe u kolokvijalnom stilu izgubila leksičku vrednost, itd

Engleski jazik, s obzirom da nema padeža zna da buda opterećen prepozicijskim frazama (tipičan primer su „of" fraze) *Grammatik* i o tome vodi računa, pa upozoreve ukoliko ih ima previše, napominjući u komentaru da njihovo gomilanje doprinosi težem razumevanju teksta, a stil lišava živosti

Psaw je gramatička struktura koja se u engleskom jeziku koristi mnogo više nego u našem, ali joj *Grammatik* nije baš naklonjan – ili pak želi da upozori da sklonost engleskog pasivnom obliku na treba shvatiti bukvalno Zato pri nalesku na pasivnu konstrukciju savetuje da razmislimo da li bi rečenica možde zvučale logičnije u aktivu Ipak je u ovoj vrsti čistunstva *Grammatik* malo preterao

Zanimljivi su statistički podaci koje *Grammatik 5* generiše o pročitanom tekstu – dobićete procenu „težine" vašeg stila – koliko je čitljiv, i koji nivo obrazovanja zahteva od čitaoca, koliko ima paragrafa i rečenica, koja im je prosečne dužina. koliko ste reči upotrebili, koja je njihova prosečna dužine

Na osnovu golih statističkih podataka, *Grammatik 5* poredi vaš tekst es tri savim različita stila angloasaksonska pisana reči prvi je čuveni govor Abrahama Linkolna održan 1863 godine u Getesburgu, posle odlučujuće bitke u amrečkom građanskom ratu (The Gettysburg Address), koji srhaičnim rečima izražava najuzvišenija patriotska osećanja drugi stilski uzorak je pripovetka Ernesta Hemingveja koja se odlikuja jezgrovitošću, kratkom rečenicom i jednostavnim rečnikom treću kategoriju predstavlje suvoparni tekst polise životnog osiguranja.

DOMETI

Mučeći godinama muku sa engleskim rečima. osetili smo izazov da ispitamo koliko ovakav jedan program može tu muku da olakša Pošteno govoreći – malo, jer se njegove pomoć još uvek svodi na sitnice. Onaj ko se lati pisanja na engleskom jeziku suočava se sa daleko većim jezičkim problemima, ali uz *Grammatik* baram neće morati da misli na lapsuse, ni u prevopisu, ni u interpunkciji, ni u osnovnim gramatičkim strukturama. Osim toga, *Grammatik 5* na elementarnom nivou može poslužiti i kao originalna alternativna metoda u učenju jezika.

Sve u svemu, imajući u vidu domete tehnologije u ovoj oblasti kao i osnovnu namenu elektronskih lektora, možemo reći da u ovom trenutku *Grammatik 5* pruža najviše što može Očekivati više značilo bi tražiti da ida ispred svoga vremena.

ADA computers

BEOGRAD
Tadeuša Košćuška72
tel/fax: 011/186-267; tel 011/186-355

NOVI SAD
Siriška 42
tel/fax: 021/416-189; 412-330 XII

386 40 MHz 4 MB 128K CACHE	1 340
486 33 MHz, 4 MB 256K CACHE VLB	1 990
486 50 MHz 4 MB 256K CACHE VLB	2 180
486 66 MHz 4 MB 256K CACHE VLB	2 330

Osnovne konfiguracije sadrže
40 Mb HDD flopi disk 1 2 Mb tastaturu mini tower kućište monitor mono herkules 14" crno beli 2S/1P port

DOPLATE

HARD DISK 120 Mb	160
HARD DISK 170 Mb	190
HARD DISK 250 Mb	290
HARD DISK 340 Mb	450
HARD DISK 520 Mb	970
VGA MONO MONITOR 14 SA VGA KARTICOM	80
KOLOR MONITOR SVGA 1024 X 768 14" 512 kb	360
KOLOR MONITOR SVGA 1024 X 768 14 1 Mb	400
VLB 1MB KARTICA KOLOR MONITOR SVGA	510

NOTEBOOK

386SX/25/2MB /80 MB HDD/FDD 1 44/MONO VGA	2850

OSTALA OPREMA

EPSON LX-400/800	410
EPSON LX 100	480
EPSON LQ-100	580
EPSON LQ-570+	840
EPSON FX-1170	1050
EPSON LQ-1070	1180
LASERSKI ŠTAMPAČ HP IV L	1670
LASERSKI ŠTAMPAČ HP IV	3490
YU SET ZA LX MODELE	30
KUĆIŠTE MINI TOWER	120
KUĆIŠTE MIDI TOWER	250
MATIČNA PLOČA 386-40 CACHE 128K	260
MATIČNA PLOČA 486-33 CACHE 256K VLB	910
MATIČNA PLOČA 486 50 GACHE 256K VLB	1100
MATIČNA PLOČA 486-66 CACHE 256K VLB	1250
PENTIUM 60 MHZ, CACHE 512K VLB, 16MB RAM	6500
I/O KARTICA 1PAR/2SER	30
VGA 16 BITNA 512K KARTICA	90
VGA 16 BITNA 1Mb KARTICA	130
VESA LB 1MB KARTICA CIRRUS LOGIC 5428	240
VLB CACHE KONTROLER 0KB CACHE	420
ISA CACHE KONTROLER 0KB CACHE	310
KOMBI KONTROLER	60
1Mb RAM MODUL	85
256K RAM MODUL	25
TASTATURA KLIK 101	60
FLOPI DISK 1,2 Mb 5 25"	110
FLOPI DISK 1 44 Mb 3 5 "	100
HARD DISK 120 Mb	380
HARD DISK 170 Mb	410
HARD DISK 250 Mb	510
HARD DISK 340 Mb	650
HARD DISK 520 Mb	1190
MONO MONITOR (Herkules)	220
VGA MONO MONITOR	240
KOLOR MONITOR SVGA 14"	590
KOLOR MONITOR SVGA PHILIPS 17"	1990
KOLOR MONITOR SVGA PHILIPS 20"	2990
MIŠ SA PODNOŽJEM	40
ETHERNET KARTICA 16 BIT	140
BNC KONEKTOR TERMINATOR	10
STRIMER CONNER INTERNI 250 Mb	550
MODEM INTERNI 2400	130
FAX/MODEM INTERNI 2400/9600	190
FAX/MODEM INTERNI 2400/9600 MNP	210
FAX/MODEM POCKET 2400/9600 MNP5/V42bis	350
MUZIČKA KARTICA SB16 AISP	400
MULTIMEDIA KOMPLET (CD ROM SB16)	940
UPS 500VA	600
KOPROCESOR 387/40	150
TONER ZA HPII / HPIIP	250/230
TRAKE ZA STRIMER	80
FILTER ZA MONITOR	30

RADNO VREME OD 9 DO 17 ČASOVA, ISPORUKA ODMAH PO UPLATI
CENE SU U DEM, DINARSKA UPLATA PO DNEVNOM KURSU NA TRŽIŠTU U BEOGRADU
GARANCIJA 12 MESECI, SERVIS OBEZBEĐEN

PC Plus

Velika usluga za malog korisnika

PC plus spada u onu kategoriju firmi gde je mali kupac u središtu pažnje. Njemu je sve podređeno – od izbora konfiguracije računara, preko saveta u svako doba, do tehničke pomoći. De zna šta je takvom kupcu potrebno PC Plus je pokazao i u našoj tek završenoj nagradnoj igri, poklanjajući upravo konfiguraciju „za svaku kuću".

Na svakom tržištu računara, pa i našem, poseban značaj imaju firme koje su okrenute malim, individualnim kupcima. PC Plus je preduzeće upravo takvog profila – specijalizovano da udovolji različitim željama širokog kruga korisnika.

Sve to ne treba da vas čudi. Ako se vratimo par godina unazad, otkrićemo tesnu vezu između PC Cluba i PC Plusa. Naime, kada je gospodin **Dušan Veljković** pre četiri godine osnovao PC Club imao je želju da stvori organizaciju koja će pomoći individualnim korisnicima računara da svoje malime što više i što bolje iskoriste. Dva godine kasnije početna ideja je dobila na kvalitetu osnivanjem PC Plusa.

„U početku PC Plus je bio sasvim drugačijeg karaktera" upoznaje nas gospodin **Donat Greber**. „Bavili smo se veleprodajom, odnosno nabavkom hardverskih komponenti sa mnogobrojne prodavce računara kod nas. All...vremenom smo težište poslovanja usmerili ka krajnjem kupcu, tako da danas PC Plus najviše radi na prodaji kompletnih računara"

U brizi sa malog kupca: Ekipa firme PC Plus - Donat Greber, Katarina Mitrović, Dragan Avramović, Zoran Kramanović, Dragan Vukelić, Zoran Rašić i Boljana Simenovski

I baš zato nam se čini da su ljudi u PC Plusu dobri sagovornici za razgovor o tipičnom domaćem individualnom korisniku računara. Prema njihovom saznanjima radi se pre svega o mladim ljudima sa izvesnim iskustvom kada su računari u pitanju. U PC Plusu oko 30 posto kupaca je ranije imalo neki personalni kompjuter

„Kupci poput naših ne dozvoljavaju nama prodavcima da se ni za trenutak opustimo. Uvek moramo biti u trendu jer naš kupac tačno zna šta želi! Kao ilustracija može da poslužiti podatak da u poslednja tri meseca ugledvnom prodajemo računare sa 486 procesorima. Tu i imamo neko kupi mašinu sa 386DX čipom dok smo matične ploče sa 386SX procesorom potpuno povukli iz prodajnog programa," ističe gospodin Greber

Tako je danas u PC Plusu najslabiji računar zasnovan na 386DX čipu koji radi na 40 MHz sa diskom od 120 MB SVGA video podsistemom i monitorom u boji. Tu je, naravno i cela paleta 486 modela. Ako neko želi još više u PC Plusu kažu da mogu da ispune gotovo svaku želju kupca. Ne zaliha imaju veliki izbor komponenti, dok je dotok novih odlično organizovan tako da ni na veoma egzotičan deo ne treba čekati duže od petnaest dana.

Stručnjaci PC Plusa će vam pomoći da izaberete optimalnu konfiguraciju računara, pružiti savet o softveru sa kojim ćete najlakše uraditi određeni posao. I na kraju ako stvari krenu naopako, na raspolaganju je dobro opremljeni servis.

Kupcima računara PC Plus ne zatvara vrata. Oduvek su se trudili da nađu vremena za njih da ih posavetuju, pomognu ili jednostavno malo porazgovaraju. Sada je sve to lakše. Jer je u okviru kuće počeo sa radom BBS, nastao u saradnji sa Radio Indeksom. Za sada BBS radi u vremenu od 17 do 7 sati ali u PC Plusu se nadaju da će uskoro početi sa celodnevnim radom. Problem je bilo obezbeđenje kvalitetne telefonske linije, no kako je nadavno puštena u rad nova digitalna telefonska centrala, to bi tebalo da se uskoro reši. Recimo još da se BBS-u može pristupiti pozivanjem telefonskog broja 011/340-227 ili preko SET Net mreže BBS-ova.

PC Plus je firma koja svakim danom želi da ponudi nešto više. Tako je nedavno u okviru kuće osnovan tim čiji je zadatak otkrivanje novog virusnog i poslovnog imidža firme. Gospoda **Dragan Avramović** čovek zadužen za marketing u PC Plusu i **Ljubiša Urošević** akademski slikar i konsultant firme paralelno se radom na ovom projektu otkrivaju i nove prostore u okviru kompjuterskog dizajna. Na kraju multimedija više nije pomodarstvo već dokazani i proveren trend koji će i kod nas uskoro uzeti maha, a za to se treba na vreme pripremiti

Sa središtem koji u središte postavlja klijenika usluga, PC Plus je uspeo da nađe svoje mesto na najboljem ali i najtežem delu tržišta računarske opreme.

Milan Bašić

Saga

Priča o SAGI

Među kompanijama na našem računarskom tržištu čiji cilj nije brza zarada već dugotrajno građenje poslovnog imidža, SAGA se posebno ističe u oblasti projektovanja i instalacije integrisanih informacionih sistema. Jedan takav veliki poduhvat odveo nas je u posetu ovoj dinamičnoj firmi, koja na krije svoje velike ambicije.

Pre četiri godine novembra 1989. u Beogradu je osnovana SAGA, tada mala kompanija ali sa velikim ambicijama da opstane na računarskom tržištu i zaista, danas u beogradskom centru „Sava", gde je sedište SAGE, radi 26 ljudi od kojih je dvanaest inženjera.

„Od samog početka SAGA je svoju poslovnu politiku zasnivala na radu sa velikim korisnicima računarskih sistema," upoznaje nas gospodin **Igor Pavlica** tehnički direktor „Doduše ni kividualne kupce nismo zaboravljali, mada nisu bili osnova našeg razvoja. Ovakav poslovni fazis nam je omogućio da isistiramo na kvalitetnim hardverskim komponentama koje ugrađujemo u računare i, uopšte, u integrisane računarske sisteme "

Naši sagovornici kažu da oni nisu prodavci računara već integratori sistema. Drugim rečima velika pažnja posvećena je razvoju računarskih mreža. U početku to su bile Novell-ove lokalne mreže kasnije je ne rad došlo „umrežavanje mreža" a danas se SAGA bavi i kompleksnim mrežnim sistemima koji premošćuju već nastalajne

„Mi računara pravimo prema potrebama kupca, upravo tako da daju mak simalne rezultate u okviru jednog sistema. Te se odnosi na celokupni program od radnih stanica pa do *high-end* servera" kaže nam **Goran Đaković** generalni direktor SAGE i dodaje „Tekvom politikom stekli smo značajne korisnike naših usluga. Rafinerju nafte Pančevo, Progres iz Beograda, Jugoxirvo, Bel Pagette i druge"

U vreme kada smo posetili SAGU pri kraju je bila instalacija velikog informacionog sistema u „Progresu". Radi se o WAN-u (*Wide Area Network*) mreži koje se prostire po celom Beogradu. Da budemo precizniji ne nekoliko mesta u gradu postavljene su lokalne mreže koje su, opet, povezane u jedinstvenu celinu. Da bi sve radilo kako treba upotrebljena su savremena dostignuća računarske tehnologije. Gde god je to moguće poslavljeni su optički kablovi a modemi rade brzinom od 64 kbauda.

Nije samo projektovanja i instalira nje sistema zaštitni znak ove firme – to su i distributerski programi SAGA je distributer za Jugoslaviju „Star" štampača, uradaja za neprekidno napajanje kompanije „American Power Conversion" i „Athana" magnetnih medija

Štampači Star su najinteresantniji distributerski program. Radi se o zaista kvalitetnim uređajima rađenim u Japanu a u SAGINOJ ponudi su modeli od najmanjih mašinčih do veoma interesantnih laserskih

Druga distributerski program – UPS uređaj – direkino je vezan za osnovnu delatnost SAGE. Bez njih se nijedan dobro projektovan sistem ne može zamisliti, a treba da budu maksimalno pouzdani. Zato se SAGA opredelila za proizvode „sa imenom", napravljene u Americi

Athana magnetni mediji obuhvataju veoma obiman program – gotovo sve što se u ovoj oblasti nudi. Da se radi o kvalitetnim proizvodima najbolje govore garantni uslovi za diskete je garancije doživotna, za strimer trake deset godina, a za magnetne trake fabrika i SAGA kao distributer daju garanciju od čak 25 godina.

Pre tri meseca osnovana je kompanija SAGA InfoTech čiji je prvenstveni cilj razvoj modernih softverskih tehnologija. Ključni ljudi u ovoj firmi su gospoda **Nebojša Vukosavović** i **Milan Dragonić** naučnici sa dugogodišnjim iskustvom u radu na projektima sa najvišim svetskim standardima.

„SAGA InfoTech je nastala u želji da se našem hardveru obezbedi i odgovarajući softver. Namera nam nije bila brza zarada, već razvoj kvalitetnog softvera koji je osnova za uvođenje pune informacione tehnologije u poslovni život preduzeća " kaže gospodin **Đaković** i dodaje: „Zato smo u SAGA Info Tech-u odabrali *ORACLE* Case jedan od najkvalitetnijih alata se ova primene."

SAGA više ne razmišlja samo u domenu PC računara i lokalnih mreža. Sada su tu i računari nadaleko čuvene kompanije Sun Microsystems kao logična posledica razvoje računarskog tržišta. gda se sve više pojavljuju korisnici kojima su neophodni data centri i radne stanice zasnovane na RISC mikroprocesorima.

Vreme u kome živimo je izuzetno teško za razvoj savremenih računarskih tehnologija. Stoga se napori preduzeća poput SAGE čine još većim i značajnijim. Ostali što bliže svetskim trendovima u neku ruku je olakšao na kome treba graditi lepšu budućnost koje možda i nije tako daleko kako nam se ponekad čini

Milan Bašić

MR SYSTEMS

MicroElectronics & Radio Systems

***Telefon* 659-506 *Fax* 687-285**

RAČUNARSKI SISTEMI

Mini tower Case 200W/LED Display
4 MB RAM SIMM 70 ns
Hard disk WD CAVIAR 220 Mb
IDE Kontroler 2S/1P/1 Game port
Floppy Disk Drive 1.44 MB TEAC
SVGA Adapter 1024 x 768, 1Mb
Color SVGA Monitor 0.28 dpi
Tastatura

386DX/40	*128Kb Cache*	**2.240 DEM**
486DX/50	*256Kb Cache*	**3.050 DEM**
486DX2/66	*256Kb Cache*	**3.250 DEM**

HYUNDAI A3/9-PINSKI

CENTRONIX CABL/YU SET

420 DEM

NOVELL/UNIX KOMPONENTE

ETHERNET KABL RG 58U 1m	**2 DEM**
ETHERNET CARD NE 2000	**150 DEM**
ETHERNET BNC KONEKTOR	**10 DEM**

MULTIPLEKSERI
TERMINAL I PRINTER SERVERI

MR BBS

18-08h 2400-14400bps/V.42bis/MNP5

tel 681-949

Kompletna hardverska i sofverska ponuda
Najnoviji **shareware** programi
DOS/NOVELL/UNIX

MT COMPUTERS
Preduzeće „MZ" 11000 Beograd
Dubljanska br. 70 Tel. 011/434-812
Fax 011 450 471

ŠTAMPAČI
EPSON

LX-400 9-pin A4 format 180 z/s ćir/lat
LX 100 9-pin A4 format 240 z/s, CSF ćir/lat
FX-870 9 pin A4 format 380 z/s ćir/lat
LX-1050 9-pin A3 format 200 z/s ćir/lat
FX 1170, 9-pin A3 format 380 z/s ćir/lat
LQ-100 24-pin A4 format, 200 z/s CSF ćir/lat ESC P 2
LQ-570+24-pin A4 format 240 z/s ćir/lat ESC P-2
LQ-870 24-pin A4 format 330 z/a ćir/lat ESC P 2
LQ 860 24-pin A4 format 300 z/s YU set kolor
LQ-1070 24 pin A3 format, 240 z/s ćir/lat ESC P 2
LQ-1170 24-pin A3 formal 330 z/s ćir/lat ESC P 2
LQ-1060 24 pin A3 format 300 z/s ćir/lat kolor
DLQ-2000 24-pin A3 format 270 z/s, YU set kolor
DFX 5000 9-pin A3 format 533 z/s ćir/lat
DFX-8000 18-pin A3 format 1066 z/s ćir/lat
SQ-870 48 mlaz A4 format, 660 z/a YU set, INK JET
SQ 1170 48 mlaz A3 format, 660 z/s YU set, INK JET
Stylus-800 48 mlaz A4 format 300 z/s CSF YU set INK-JET
Stylus-1000 48 mlaz A3 formal 300 z/s CSF YU set INK-JET
EPL 5200 Laser 300 dp 6
s/m 2MB PCL4/PCL5 ESC/9 ESC/24 YU-set
GT-6500 SCSI color scannar A4 format 600 DPI
GT-8000 COLOR IMAGE SCANER A4 formal 600 DPI

Roland DIGITAL GROUP

DXY 1100 A3 format 8 pera 42 cm/s mag držanja papira
DXY-1200 A3 format 8 pera 42 cm/s ele statičko
DXY 1300 1MB A3 formal 8 pera 42 cm/s ele statičkor
SKETCH MATE A3 format, 8 pera
SKETCH MATE A4 format, 8 pera
DPX-2500 A2 plotar tabla 8 pera, 62 cm/s 1 MB
DPX 3500 A1 plotar-tabla 8 pera 62 cm/s 1 MB
GSX-3000 A1 „ROLL"-plotar 8 pera 113 cm/s 1 MB aut p v p
DPX-4600 AO plotar tabla 8 pera 62 cm/s 1 MB
GSX-4010 AO „ROLL" plotar 8 pera 113 cm/s 1 MB aut p v p.
LTX 100 A3 format termički ploter
LTX 120 A3 format, termički ploter
LTX 321 A1 format termički ploter
LTX 420 A0 format termički plotar
Data buffer SYA 550
CAMM-1 PNC-1100 PLOTER-REZAČ formal od 50 do 640 mm (do 1600 mm)
CAMM-1 PNC-1800 PLOTER-REZAČ A0 format (do 3600 mm)
STIKA Rezač sa ugrađenim skenerom se RS232C r programom

EIZO

EIZO 9080i 16 kolor monitor 1024x768
EIZO 6500 21" monohromatski monitor
1664x1200(60Hz)1280x1 024
EIZO F550i 17" kolor mon ravan ekran max. 1280x1024
EIZO F750i 21" kolor mon ravan ekran max 1280x1024
EIZO T560i 17" kolor (TRINITRON CRT) 1280x1024 CRT Trio p 0.3I
EIZO T660I 20" kolor (TRINITRON CRT) 1280x1024
EIZO GRAFIČKE KARTICE — VA30 VA41 AA51 MD 809 MD B10 MD B12

GARANCIJA GODINU DANA
POSEDUJEMO DODATNI I POTROŠNI MATERIJAL,
RIBONI PERA (ROLAND)
ĆIRILIČNO-LATINIČNI EPROMI za LX FX, LO MODELE
********* OVLAŠĆEN SERVIS FIRME — EPSON, ROLAND ***********
INFORMACIJE NA TEL 011 434 812 FAX 011 450 471

OLYMP electronic

11000 BEOGRAD
Jovana Đaje 10
tel. 011/400-477
fax 011/410-240

EPSON
EIZO
Roland DIGITAL GROUP

386-33 SX
RAM 1Mb
FLOPPY 5.25"
HD 42Mb
IDE CONTROLER 16bit
VGA 512Kb
SVGA MONO MONITOR 14"
DESKTOP CASE
TASTATURA 101
1090

386-40 DX
RAM 4Mb 64Kb CACHE
FLOPPY 5.25"
HD 126Mb
IDE CONTROLER 16bit
SVGA 512Mb
SVGA MONO MONITOR 14"
DESKTOP CASE
TASTATURA 101
1730

486-33 LOCAL BUS
RAM 4Mb 256Kb CACHE
FLOPPY 5.25"
HD 129Mb
IDE CONTROLER 16bit
SVGA 1Mb
SVGA COLOR MONITOR 14"
MINI TOWER CASE
TASTATURA 101 MIŠ
2720

486-50 LOCAL BUS
RAM 4Mb 512Kb CACHE
FLOPPY 5.25"
HD 210Mb
VL-BUS CONTROLER 32bit
SVGA 1Mb VL-B 16.7 mil. boja
SVGA COLOR MONITOR 14
MINI TOWER CASE
TASTATURA 101 MIŠ
3370

486-66 EISA
RAM 8Mb 512Kb CACHE
FLOPPY 5.25" i 3.5"
HD 540Mb SCSI
SCSI EISA CONTR. Cach 4Mb
SVGA 1Mb 16.7 mil boja
SVGA COLOR MONITOR 14"
LARGE TOWER CASE
TASTATURA 101 MIŠ
7490

Mainboard

MB-386/33 SX	190
MB-386/40 DX	310
MB-486/33 LOCAL BUS	990
MB-486/50 LOCAL BUS	1250
MB-486/66 LOCAL BUS	1450

SVGA

SVGA mono monitor	250
SVGA color monitor	590
SVGA 1Mb	150
SVGA 1Mb 16,7 mil boja	180
SVGA VL-BUS 16.7 mil. boja	270

Diskovi

HD-85Mb	350
HD-120Mb	430
HD-210Mb	560
HD-320Mb	700
HD-540Mb	1450

Komponente

Floppy disk 5.25"	130
Floppy disk 3.5"	120
Memorija SIMM 1Mb	100
16bit IDE controler	50
32bit VL-BUS controler	150

Dodatna oprema

Ethernet card 16 bit	190
Miš	50
Filter stakleni /14-18/"	140
Ribon za A4/A3	20/30
Centronics kabl	20

Kućišta

DeskTOP	160
Slim	200
Mini TOWER	200
TOWER	240
Large TOWER	300

EPSON

Printeri 9.pina

LX-400	440
FX870	1080
FX-1170	1180
LX-100	540
DFX-5000	3980

Printeri 24.pina

LQ-100	640
LQ-570	880
LQ-1070	1280
LQ-870	1480
LQ 1170	1880

INK-JET

STYLIS-800	1100

SCANERS

GT-6000	4200
GT-8000	5100

Roland

Ploteri A3

DXY-1100	1780
DXY 1200	2210
DXY-1300	CALL

Ploteri A2

DPX-2500	CALL

Ploteri A1

GSX-4000	CALL
DPX-4600	CALL

Ploteri A0

GSX-4000	CALL
DPX-4600	CALL

Data Buffer

SYA-550	1650

EIZO

Monitori

6500	21"m	2900
F340I	15"	1950
F550I	17"	2750
T560i	17"	CALL
T660i	20"	CALL

Printeri

LASER JET IV L	1900
LASER JET IV	3490
LASER JET IV M	5900
DESK JET 500C	1390

HARDVER

Procesori

AMD: Windowe logo i novi čipovi

Microsoft je dozvolio da novi AMD-ovi 80386 i 80486 kompatibilni procesori ponesu *Windows* logo istovremeno jedan od najvećih Intelovih konkurenata predstavio je dva nova čipa koja kloniraju 486-icu

Prvi je mikroprocesor sa radnim taktom od 40 MHz koji u sebi ima i deo zadužen za upravljanje potrošnjom električne energije dok je drugi AMD-ove verzija 486DX2 čipa sa istamim taktom od 66 MHz Sporiji procesor namenjen je tržištu prenosnih računara i stonih „telenih" modela gde je poseiina pažnje posvećena uštedi struje Što se tiče cene ona će biti približno jednaka Intelovom 486 čipu koji radi na 33 MHz

Brzi Am486DX2-66 procesor po rečima predstavnika fabrike je pre svega, „alternativni izvor" za proizvođače kompjutera Drugim rečima AMD očekuje da će se neke od fabrika opredeliti da od sada 486DX2 čip ne kupuju više kod Intela Na kraju ako vas interesuju cena sa količinu od hiljadu komada evo ih 40 MHz čip u PGA kućištu košta 283 dolara, a za DX2 u istom kućištu treba platiti 463 dolara

Kontakt adrese. AMD, tel 408-749-2618

Procesori

Cyrix: Še 8386SX na 486 o jednom potezu

Evo i jedne vesti koja bi mogla da obraduje mnoge vlasnike PC računara kod nas Cyrix Corp. je predstavila novi mikroprocesor sa oznakom Cx486SRe2 koji omogućava vlasnicima računara se 386SX procesorom na radnom taktu od 16 20 ili 25 MHz da iako i relativno jeftino postanu ponosni vlasnici mašine čije performanse pripadaju 486 klasi

U Cyrix-u kažu da novi čip ima kolo za udvostručavanje takta jeden kilobajt internog keš memorije poboljšane ulazno-izlazne operacije nad podacima izvršavanja instrukcija u jednom ciklusu i u hardveru implementiranu jedinicu za množenje i inkrem

Da bi instalacija bila što jednostavnija Cyrix je angažovao kompanij u August Inc koja je razvila „clip-on" sklop kojim se jednostavno zamenjuje postojeći 386SX mikroprocesor Automatski sa iskljućuje originalni čip i novi mikroprocesor preuzima sva potrebne funkcije Ovaj upgrade paket namenjen je isključivo desktop i tower računarima, jer zbog dimenzija i veće toplote nije moguće priključivanja na prenosne modele.

U kitu koji se prodaje diljem Amerike nalaze se mikroprocesor „on-clip" sklop instalaciono uputstvo i softver sa instalirenje keša Proizvođač navodi da je za ceo postupak instalacija potrebno oko 15 minuta a preporučena cena verzija na 25 MHz je 299 dolara.

Nešto pre ovog procesora Cyrix je predstavio Cx486DRe2 koji na jednostavno lak način zamenjuje i386DX procesore Filozofija je ista: u dva poteza dobijate mašinu sa procesorom koji razume specifične 486 instrukcije U inostranstvu su već napravljeni prvi upoređni testovi Cx486DRe2 procesora i odgovarajućih prvih 486-ica Prema nekim pokazateljima Cyrix-ov čip sa spoljnim taktom od 20 MHz potpuno se ravnopravno nosi sa 486SX procesorom na 25 MHz kada su u pitanju aplikacije koje rade pod DOS-om Nasuprot tome testovi rada pod *Windows*-om pokazuju laganu prednost i 486SX čipa

Ocena na testovima je bila da za ista para možete kupiti Cyrix-ov *upgrade* paket ili matičnu ploču sa 486SX mikroprocesorom. Prva varijanta vam nudi laku i brzu instalaciju. dok je druga ipak celovito rešenje koje traži iskustvo za ugradnju a ne samo „krpelj" na staroj ploči

Kontakt adrese. Cyrix Corp , tel. 800-462 9749

Memorije

Flesk kartice doleze brzinom munje

Matsushita Electronics Industry je zajedno sa kalifornijskim Sundisk-om razvila šesnaesto-megabitnu Flash memoriju Kako se saznaje početkom sledeće godine japanska kompanija će početi sa serijskom proizvodnjom čipa, koji će američanci ugrađivati u svoja memorijska kartice

Memorijski čip je proizveden u originalnoj 0,5 mikronskoj tehnologiji i predstavlja logični nastavak ri zvoja koj je počeo još 1990. godine sa četvoro i osmo bitnim Flash memorijskim čipovima.

Sundisk-ove Flash memorijske kartice u koje će biti ugrađena nova memorija biće kapaciteta 40 megabajta. To je dva puta više nego što druge firme nude na tržištu. Kapacitet je inače tako odbran da se na jednoj kartici može pohraniti Windows softver i time učiniti prihvatljivim za korisnike notebook računara

Slikla nam ja još jedna mformacija koja najavljuje novu Flash memoriju Zilog i Catalyst Semiconductor zajednički će razviti Flash memorijsku karticu Prema navodima u zvaničnom saopštenju nova memorija će imati cenu po megabajtu uporedivu sa klasičnim hard diskovima Ipak, poznavaoci prilika na ovom segmentu tržišta su skeptični Cena megabajta na hard disku daleko je kreće oko jednog dolara, dok se za Flash memoriju procanjuje da u narednih godinu dana ne treba očekivati apuštanje cene ispod 2 do 2,5 dolara po megabajtu

Japansko-američka kartica ima dvostruko veći kapacitet od konkurencije ak po svemu sudeći njihova dominacija neće dugo trajati Zilogova Flash kartica treba da bude kapaciteta od 80 do 140 megabajta

Kontakt adrese. Matsushita Electronics Industry +81 726-62 5521 Zilog, tel 408-370-8246 fax 408-370-8056, Catalyst Semiconductor tel 408 744-0260)

Video kartice

ATI: Bolji kvalitet slika

Dobro poznata kompanija, kada su u pitanju video podsistemi ATI Technologies najavila je dve nova proizvoda koja će se pojaviti u drugom kvartalu 1994 godine Reč je o ATI 68800DX Advanced Graphics and Video Accelerator-u i ATI 68890 Video Capture Processor-u

Kao i kartice 68800LX odnosne 68800AX i ATI 68800DX ima mach32 grafički kontroler ali podržava i SFBI standard poseond namenjen multimedia softveru Takode dodat je i video display processor koji poboljšava performanse prikaza slika u punom pokretu Iz ATI-je ističu informacija da će nova kartica biti sposobna da skalira sliku po vertikalnoj i horizontalnoj osi, konvertuje boje iz RGB u YUV standard i obratno kao i da podrži specijalne efekte poput mešanje video signala. Predviđeno je da će se ATI 68800DX proizvoditi u verzijama za priključenje na ISA, VESA ili PCI magistralu

ATI 68890 je namenjen pojedinačnim ili integrisanim video podsistemima i kako kaža proizvođač može „hvatati" NTSC ili PAL sliku bez gubitka u kvalitetu

ASYS COMMERCE
tel. 021/623-928 624-501 618-887
EPSON
LQ 100 500 DEM
LQ 570 790 DEM
LQ 1070 1090 DEM
YU SET I CENTRONICS KABL

Naravno, obezbeđen je i kvalitetan transfer televizijskih standarda u RGB odnosno YUV format.

Objavljena je i cena novih proizvoda U početku dok se broj porudžbina ne poveća ATI 68800DX će koštati 90 dolara a za ATI 68890 biće potrebno potrošiti 50 dolara

Video kartice

ATI: VIDEO WONDER ispunjava vaše želje

U skladu sa trendom koji vodi ka full-motion videu ATI Technologies ja proizvod dodatnu ploču za PC po imenu VideoWonder Po rečima proizvođača Video Wonder je prva kartice kod koje je postignuta kombinacija *full-motion* videa i visokih performansi

Ploča „hvata" i komprimuje video sliku i istvarnom vremenu u jednom jedinom koraku. Prikazuje video u željenom prozoru brzinom od 30 okvira u sekundi *(full-motion)* na bilo kojoj grafičkoj rezolucji Konsrak može da reprodukuje kompromovane video sekvence sa hard-drajva na celom ekranu *full-motion* brzinom. VideoWonder se instalira i upotrebljava vesma jednostavno Nisu potrebni nikakvi kablovi za povezivanje, jer je sva grafika smeštena na ploči što opet smanjuje konflikte

Sve komponente su dizajnirane i optimizovane da rade zajedno, čime se postiže funkcionalnost i nivo performansi koji na može da pruži ni rešenje sa dva ploče niti video ploča na koju su dodate VGA komponente

Visok kvalitet video plejbeka enači da se video može prikazivati *full-motion* brzinom bez ograničenja na prvobitnu veličinu u kojoj slika snimljena Mach32 video akcelerator ima ugrađan ntpvi Video Display Processor koji omogućava skaliranje slika tako da slika deluja prirodno i nema uobičajenog trzanja.

VideoWonder daje 65 000 boja u rezolucji 1024x768 sa frekvencijom vertikalnog osvežavanje od 76 Hz. 16 7 miliona boja u razolucji do 800 x 600 te 256 boja u rezolucji 1,280x1024 se frekvencijom osvežavanja od 74Hz

Ostale pogodnosti koje nudi ova ploča su snimanje slike pod DOS-om i pod *Windows*-om animanje „zamrznutih slika" u punom broju boja, *preview* rečim za animanje video slika Ploča se isporučuje sa 2 MB video RAM (uz mogućnost konšenja još 2 MB DRAM) i sa softverom sa animanje i editovanje slike u *Windows*-u

Cena je 999 USD a prodaje treba da počne u aprilu 1994 Radiće se verzije za ISA magistralu kao i za PCI i VL bus

Netpower

RISC računari za Windowe NT

Netpower je nedavno počeo sa prodajom familije personalnih računara sa RISC mikroprocesorima koje je projektovana za rad sa *Windows NT* operativnim sistemom *RISC Series 100* računan konsta dobro poznati i provereni MIPS R4400 procesor Kako kažu u kompaniji Netpower arhitektura računara je optimizovana sa rad u mreži, a može da posluži i kao vrhunska radna stanica

Ako je verovati proizvođaču performanse su dvostruko bolja nego kod sličnog računara zasnovanog na Intelovom mikroprocesoru Osnovna konfiguracija uključuje 128 kilobajta sekundarnog keša integrisani video podsistem sa lokalnom magistralom dva slota za proširenje po ISA standardu flopi disk od 3,5 inča, šest nosača za diskove *ethernet* SCSI para lelni i dva serijska porta

Za manje od 5500 dolara u Americi možete kupiti Netpower Series 100 sa procesorom koji radi na 100 MHz, 16 MB RAM-a, diskom kapaciteta 245 MB i kolor monitorom dijagonala 14 inča. Onima sa više novca i većim zahtevima namenjen je računar čiji R4400 procesor radi na 134 MHz, ima 32 MB RAM-a, 525 megabajta na hard disku CD ROM drajv i monitor sa ekranom dijagonale 17 inča Sve to vredi nešto manje od 10 hiljada dolara. Na vrhu ponude se nalazi server

ska konfiguracija se procesorom na 150 MHz čija je cena oko 18 hiljada dolara.

Kontakt adrese. Netpower Inc. tel 800-801-0800

Diskovi

Seagate: Gigabajte oškad desto

Ako se nastavi ovakav trend rasta softverskih paketa, diskovi danas namenjeni serverima postaće premali i za obična računarska! Moramo priznati da je naša prethodna konstatacija, ipak malo preterana ali nije ni previše daleko od istine. Da je tako pokazuju izveštaji se Konferencije o masovnom memorijama velikog kapaciteta koja je na Comdex-u održana pod pokroviteljstvom Seagate-a

Osnovne teme kojima je konferencija posvećena vezane su za probleme i ograničenje postojećih arhitektua u korišćenju velikih diskova. Tako su nekada nedostižnih 528 MB na IDE kontroleru i 8 4 MB na SCSI interfejsu danas progašeni kao „niski limiti u personalnim računarima radnim stanicama i fla serverima"

Da bi postetinicima konferencije problemi bili što plastičnije prikazani obezbeđen je izvestan broj računara ranominanih proizvođača (koji, naravno koriste Seagate-ove diskove) Tu je bio i softver gledan prostora na disku poput *Windows NT-a*.

Interesantno je reći da konferencija nije donela nikakve revolucionarne zaključke Uglavnom svi učesnici su se složili sa postojanjem problema i potrebom da se oni reše Ipak to je ostavljeno da se uradi drugom prilikom I na nekom prihvatljivijem mestu nego što je to sejameka manifestacija, pa makar ona bila I – Comdex

Diskovi

Quantum: Mole je veliko

Nedavno je Quantum predstavio šest novih hard diskova koji svojim performansama i kapacitetom pokrivaju sve segmente tržišta personalnih računara – od notebook-ova do moćnih servera za lokalne mreže Radi se o četiri nove Quantum Daytona diska I dva Quantum Empire modela.

Quantum Daytona, najmlađa familija u proizvodnom programu, namenjena je najmanjim računarima, bar kada su dimenzije u pitanju. Disk kapaciteta 256 megabajta visok je samo 12,5 milimetara (pola inča) I potpuno odgovara subnotebook-ovima Tu je I model kapaciteta 514 MB ali sa visinom od 19 mm što je sasvim zgodno za prenosna računara dimenzija beležnice. Slični dimenzija su I druga dva Daytona diska ali sa kapacitetima 127 MB, odnosno 341 MB

Sada se kupcima nudi ukupno šaanaest Daytona diskova čije su zajedničke karakteristike dimenzije od 2 5 inča, brzina okretanja od 4500 obrtaja u minuti i prosečno vreme pristupa od 17 milisekundi. Treba posebno istaći nivo elektroniku na diskovima koja poboljšava ukupne performanse korišćenjem novih tehnika keširanja. Daytona siekovi se isporučuju u verzijama se SCSI-2 I IDE-AT vezničima, a tu su i modeli predviđeni za postavljanje na VL I PCI lokalne magistrale.

Kod konstruisanja ovih diskova u Quantumu su posebnu pažnju posvetili potrošnji električne energije. Prosečana potrošnja je samo 0,2 W. a to je postignuto jednim delom zahvaljujući *Autoidle* i *StackWrite* tehnikama. *Autoidle* isključuje aktuator posle pet sekundi neaktivnosti diska. *StackWrite* tehnika omogućuje brzi prelazak iz stanja mirovanja u pun pogon čima se smanjuje potrošnje struje kod standardnih aplikacija i do 10 posto, bar tako tvrde u fabrici

Druga dva nova diska pripadaju *Empire* familiji Ono što ih izdvaja od većine drugih diskova na tržištu je PRML *read channal* tehnologija, koja je znatno efikasnija od ensiogne *peak detect read chanal* tehnike Ova osobina naročito je važna kod rada u aplikacijama koje zahtevaju često čitanje podataka se diska.

PRML tehnika inače nije nova Još sedamdesetih se prvi put primenjena u digitalnim komunikacijama, ali tek 1990 godina IBM je prvi put upotrebio na hard diskovima. Korišćenjem posebne šema kodiranja, PRML povećava količinu pročitanih podataka. Sekundami pozitivni efekt je veća gustine pakovanja podataka na površinu diska, što uvećava kapacitet.

Quantum je dodatno unapredio PRML tehniku koja je primenjena na novim Empire diskovima. Tako je sada unutešnji protok podataka podignut na 64 megabita u sekundi što je za oko 50 posto bolja nego kod drugih diskova. Drugim rečima, brzina prenosa podataka je podignuta na 5 Mbps

Empire 2160 ima kapacitet od 2 1 GB formatiranog kapaciteta I prvi je disk od 3 5 inča koji je prevazišao granicu od 2 gigabajta Drugi model Empire 1440 karakteriše mala visina I kapacitet od 1 44 GB Zajedničko se oda modela je brzina okretanja od 5400 obrtaja u minuti I prosečno vreme pristupa podacima od 9,5 milisekundi Diskovi troše od 8 od 9,5 W

Prvi uzorci novih Quantumovih diskova već su kod proizvođača kompjutera, a cena im se kreće od 250 od 1495 dolara. Masovna proizvodnja predviđena je za prvo tromesečje 1994 godine

Kontakt adrese. Quantum Corporation, tel 408-894-4000

Diskovi

Maxtor: Sami zamenite disk većim

Popularnost *Windows-a* je postala tolika da mnogi korisnici računara u svetu počinju da razmišljaju o na dogradnji svojih mašina tvrdim diskovima većeg kapaciteta Pratiš ovaj trend Maxtor je odlučio da na tržište izbaci svoje diskove serije 7000 u pakovanju namenjenom ugradnji po sistemu „sam svoj majstor"

Ovi diskovi veličine 3 5 inča I kapaciteta 130 213, 245 i 345 megabajta prodaju se zajedno sa nosačem za disk, kablovima I drugim potrebnim hardverom a tu su I neophodna uputstva I instalacioni softver

Kupovinom Maxtor Series 7000 sisi za nadogradnju dobijate garanciju servis I podršku direktno od proizvođača. Cene pakela kreću se od 339 od 579 dolara

Kontakt adrese. Maxtor Corp tel. 800-482-9867

SOFTVER

Razvoj softvera

Microsoft: Fortrao za Wiodowo NT

Da fortran nikada ne gubi na aktuelnosti pokazuje i najnoviji Microsoft-ov proizvod se programere – tridesetdvobitna verzija ovog programskog jezika namenjena razvoju aplikacija koje rade pod *Windows NT* operativnim sistemom

Fortran PowerStation 32 ja prvenstveno dizajniran za tržište naučnih I inženjerskih radnih stanica, kao i za organizacije koje imaju potrebu da svoj obično veoma specifičan softver. prenesu se *mainframe* i mini kompjutera na PC računare. Takođe, u Microsoft-u očekuju da će i nezavisni projektanti softvera biti dobri kupci novog fortrana

ASYS COMMERCE
tel 021/623-928 624 501 618-887
EPSON
FX 1170 1050 DEM
DFX 5000 3900 DEM
YU SET I CENTRONICS KABL

Svi koji su donada imali priliku da se sretnu se for tranom na PC računaru znaju koliko muke može prouzrokovati segmentna arhitektura koju nameće DOS Sada *Windows NT* pruža mogućnost da se u jednom komadu adresira od četiri gigabajta memorije e to ja više nego dovoljne za bilo kakvu aplikaciju pisanu u fortranu Naravno to nije jedina nova i dobra osobina *Microsoft Fortran PowerStation 32* kompajlera Pored standardnih lokalnih I globalnih optimizacija koda tu su I specifične optimizacije za Intelove i486 i Pentium procesora kao i automatsko *inline* procesiranje, čime se smanjuje prekoračenje pozivanja procedura. Sve ovo omogućava pisanje glomaznih simulacija koje se mogu izvršavati na PC računaru

Paket, pored kompajlera sadrži *Visual Workbench*, integrisano razvojno okruženje koje je već poznato iz poslednje familije *Visual C++* programskih paketa. U Microsoftu navode da novi proces razvoje aplikacija koji nameće *Visual Workbench* skraćuje vreme rada na pisanju programa, jer je sada sve na jednom mestu tako da nema potrebe se stalnim prelaskom iz jednog alata u drugi

Da biste koristili *Fortran PowerStation 32* potreban vam je najmanje PC računar se i386 procesorom na 25 MHz 16 MB RAM-a jedan flopi drajv, 12 megabajta prostora na disku I *Windows NT*. Preporučena cena za paket je 795 dolara ali ako se na kupovinu odlučite do kraja januara 1994 godina možete uštedeti čitavi 300 dolara.

Kontakt adrese. Microsoft Corp.. tel 206-882-8080

Specijalizirani paketi

Borland i WordPerfect ozvraćaju udarac

Tokom poslednjeg Comdex-a Borland i WordPerfect su predstavili novu drugu po redu verziju paketa namenjenog kancelarijskom poslovanju *Borland Office 2 0* zamišljen je tako da može u potpunosti da se uhvati u koštac sa istoimenim Microsoftovim paketom

U jednoj kutiji pred kupce dolaze tri veoma popularna I kvalitetna programa radna tabela *Quatro Pro 5 0 for Windows* baza podataka *Paradox 4 5 for Windows* I poslednja verzija najpopularnijeg tekst procesora na svetu *WordPerfect 6 0* takođa u verziji se rad pod „prozorima" Na predstavljanju paketa u Las Vegasu zvaničnici obe kompanije na sav glas su ukazali da je njihov proizvod sastavljen od dakova koji su, sami za sebe, kod kupaca i kritika već pobrali vrhunske ocene.

Ipak hvalospevi nisu jedino što spaja ove tri aplikacije Tu je I nova instalaciona procedura koja automatizuje proces za sva tri programa. Čime konanik od prvog trenutka stiče utisak da se radi o jednom integrisanom sistemu moćnih odlika

Istovremeno se među prisutnim novinarima na prezentaciji javilo I pitanje da li se to možda aprama ujedinjenje Borlanda i WordPerfecta. Alan Elton (Alan Ashton) predsednik WordPerfect Corp jasno je stavio do znanja da se ujedinjenje na planira ali će se na prodaji i unapređenju *Borland Office 2 0* paketa veoma tesno sarađivati Filip Kan (Phillipa Kahn) prvi čovek Borlanda je bio još jasniji Za njega saradnja dva kompanije odvijače se jedino na širiji ovog proizvoda, ali u punoj meri što uključuje I zajednički marketinške nastupe organizovanje seminara kurseva I svega ostalog što u svetu prati uspešne proizvoda

Prodaja paketa *Borland Office 2 0* počeće u decembru a preporučena cena je 595 dolara. Korisnici ranija verzije ili sličnih paketa drugih proizvođača se prelazak na novu verziju traba da plata 295 dolara.

Kontakt adrese. Wordperfect Corp , tel 800-331-0877

Tekst procesori

WordPerlect: Ceo disk za pisanje

Ako na disku imate slobodnih 33 megabajta i obožavate *WordPerfect* a bez *Windows-a* ne možete zamisliti život uz računar se vas je prava stvar novi *WordPerfect 6.0 for Win* ili ako vas impresivaju brojke za 312 dana jedan od najpopularnijih tekst procesora

za Windowsa-e je narastao za oko 21 megabajt, što znači da je svakog dana dobijao 69 kilobajta, odnosno svakog minuta prostor potreban na disku postajao je veći za 48 bajta. Zašto, da li sve to ima ikakvog smisla?

Naravno, u WordPerfect-u smatraju da glomaznost poslednje verzije tekst procesora nije uzaludno bačen prostor na disku. Proizvođač kaže da je korisnički interfejs sada čišći i više grafički orijentisan, a to je zahtevalo pisanje potpuno novog koda. Na vrhu ekrana, pored menija je linija se ikonama preko kojih možete pozvati najčešće korišćene funkcije. Ništa novo, reći ćete, ali ovog puta sve radi brže i jednostavnije, a zamislite ima čak 81 ikonica. Ono što je na prvi pogled nove je to da u prvoj liniji na ekranu, umesto imena programa i datoteke koju obrađujete, možete videti help liniju.

Ipak, 33 megabajta nije potrošeno samo na korisnički interfejs. Prema prvim informacijama koje su do nas stigle *WordPerfect 6.0 for Windows* je proizvod koji je još više smanjio razliku između tekst procesora i DTP aplikacija. Tu su vam sada, na primer i mogućnosti formatiranja teksta koje uključuju Bezijerove krive i slične stvari. Ni poboljšanja u formatiranju tabela nisu se zanemarivanje. Sada se pronađen podatka u tabeli možete koristiti se formula – pravi *spreadsheet* u tekst procesoru.

Antivirus programi

Borba protiv virusa: Baliko ama bezbodni

Prema poslednjem istraživanju britanskog *Virus Bulletin*, publikacije namenjene zaštiti od kompjuterskih virusa, neki od najpopularnijih programa se otkrivanja nepoznati koje mogu uništiti vaš računar i nisu toliko bezbedni kolko ih proizvođači hvale. Posebno loše ocene dobili su rezidentni (TSR) moduli.

Microsoftov program koji se isporučuje uz DOS 6.0 prošao je najgore na testu, uspeo je da otkrije samo 58,4 posto virusa. Nešto bolja rezultata postigli su McAfee-ov Central Point-ov i Frisk-ov F-PROT rezidentni programi sa 78,8 posto uhvaćenih virusa.

Ieti programi, kada nisu postavljeni u mod za skeniranje „napeanika" već im je prepušteno da ispituju disketu se zaraženim programima pokazali su daleko bolje rezultate. Kod nas veoma popularni *VSHIELD* je otkrio 66,8 posto poznatih virusa. Slično su prošli i drugi testirani programi a najlošije rezultate je pokazao RG Software-ov *Vi-Spy* sa stoprocentnim u činkom. Pouka koja se sama nameće je ne verujte TSR programima, radije odvojite malo vremena za skeniranje diskete. Time će zaštita vašeg računara biti da teko bolja.

Alatke

Norton Desktop: Pa treći put a Prazarima

Nema sumnje da su Nortonove alatke najpopularniji paketi se održavanje „higijene" na računarima. Pored verzija se DOS, u poslednje vreme sve više je korisnika koji se odlučuju da pređu na Windows verziju koja je umesto imena *Norton Utilities* nazvana *Norton Desktop* i sada je dobila svoje treće izdanje.

Red je da vam predstavimo poboljšanja u odnosu na prethodnu verziju. Dodati su novi alati *FileAssist*, *Day Planer* i *Script Maker*. Tu su još novi *on-line* tutorijal kao i help. Za one koji su navikli da koriste *Control Center* napisana je verzija koja radi pod *Windows*-em.

U informacijama koju je Symantec pravadio prilikom predstavljanja paketa piše da su unapređeni moduli se dijagnostiku i oporavak oštećenih datoteka, bekap i otkrivanje virusa. Tako sada *Norton Disk Doctor* može da otkloni probleme bez izlaska u DOS ieto važi i se *Speed Disk* modul.

Pomenuli smo nove module, a sada i reč-dve o njima. *FileAssist* omogućava lakši red se datotekama i podržava mogućnost da se datotekama daju imena duža od osam slova. *Day Planer* je dobra zamena za Microsoftov *Calendar*.

Norton Anti Virus 3.0, prema rečima proizvođača sada otkriva 2300 virusa, a poseduje i mogućnost otkrivanje nepoznatog koda koji može da ošteti vaše podatke. *Norton Backup 3.0 for Windows* ima poboljšan interfejs se kojim se i neiskusan korisnik može lako služiti. Povećan je i broj medija koji se koriste za bekap. Sada su tu jedinice trake koja rade pod mrežom, kao i bekap uređaji se SCSI vezivkom.

O radu pod nekom od mreža na treba trošiti puno reči. Zar još postoji neki renominani softverski paket koji ne podržava *Novel NetWare*, *Banyan Vines*, *Microsoft LAN Manager* ili *LANtastic*. Cena paketa je 179 dolara, a za prelazak se stare verzije registrovani korisnici treba da plate 40 dolara.

Kontakt adresa: Symantec, tel. 310-449-4140.

Fontovi

Zbirka Adoba fontava na CD-ROM-u

Adobe Systems najavljuje verziju *Adobe Font Folio 6.0* na CD-ROM-u koja sadrži više od 1.000 Type 1 slovnih likova za Macintosh i PC računare. Po prvi put korisnici softvera *Adobe Font Folio* mogu da pristupe bibliotaci *Adobe TypeLibrary* na jednom disku. bilo u Macintosh ili *Windows* okruženju.

Nova verzija nudi slovne likove od 1 do 360 iz *Adobe Type* biblioteke, *Adobe Acrobat Reader* se pregled dokumenta, kao i štampanja bilo kojeg dokumenta u novom Adobe formatu, PDF (*Portable Document Format*).

CD-ROM pruža mogućnost izlaza na bilo koja dva izlazna uređaja a sadrži i sledeće softverske proizvode: *Adobe Type Manager*, *Adobe Type Reunion* (za Macintosh), slovne likove (*Multiple master* (za Macintosh), *Adobe TypeAlign*, *Adobe Wild Type* fontove *Adobe Acrobat Reader*.

Softver po imenu *FontMinder* služi za organizovanje fontova na PC računaru, zajedno se dokumentacijom i tehničkim podacima. Preporučena cena je 14.000 USD do 31. marta 1994 (kasnija 17.000).

ZANIMLJIVOSTI

Bizarni softver

Bajzbol timovi čuvaju ekran

Quadrangle Software je nedavno predstavio *Screen Saver* koji je dizajniran tako da se svidi zagrižienim ljubiteljima sporta. Time se kolekcija bizarnih čuvara ekrana, poput već čuvenih „letećih krava" upotpunila emblemima

najpopularnijih američkih bejzbol klubova i atraktivnim prizorima ove igre koja u Evropi nikada nije stekla popularnost.

Bejzbol čuvar ekrana omogućava korisniku da izabere jedan od deset planova koji prikazuju različite pokretne slike i proizvode tipične zvuke za ovaj sport. Takođe, korisnik bira „domaći" i „gostujući" tim iz palete od 26 timova koji se takmiče u Američkoj nacionalnoj ligi.

Kao i kod svakog drugog „oživljnog" programa, tu je i niz opcija za podešavanje, poput brzine odvijanja akcije, poruka koje se ispisuju na ekranu i drugo.

Sajmovi

Kompjuterski magavi u vazdušnoj borbi

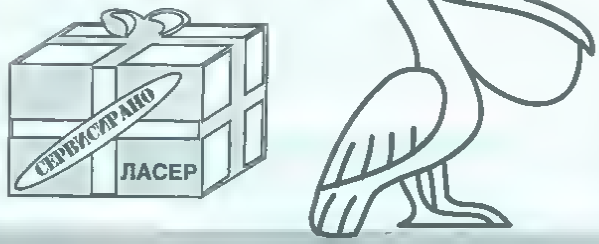
Tokom ovogodišnjeg Comdex-a biće organizovana prava vazdušna bitka u kojoj će učestvovati pedesetak šefova kompjuterskih kompanija koji poseduju pilotsku dozvolu. Među prijavljenim imenima su i takve veličine kao Filip Kan (Borland) i Ed Eliber (Creative Labs.)

Nad pustinjom u okolini Las Vegasa svoje pilotske umeće će ukrstiti na avionima Marchesti SF 260. Cela program zbog bezbednosti uključuće trening i borbeno letenje. Sudija meča biće Majk Bleston osnivač Air Combat USA, kompanije koja organizuje razne svijetlarske manifestacije. Sva akcija tokom borbe biće zabeležene na video traci, a slike će beležiti kamera postavljena umesto pravog mitraljeza.

Da bi utisak bitke bio što realniji avioni će biti opremljeni posebnim radio sistemom koji kada je aparatu „pogođen" aktivira dimni signal. Ishod ovog „rata" kompjuterskih magova nije nam poznat ali atraktivnost manifestacije sasvim sigurno ističe prave pobednike – organizatore Comdex i Air Combat USA.

IMTEL computers
Super VGA - 512k
MONO MONITOR
MIS
DOPLATE:
386SX / 120 MB=1230
1 MB , 144 MB
COLOR MONITOR +330
386DX / 170 MB=1450
2 MB 144 MB
HDD 170MB +50
HDD 250MB +120
486DX / 170 MB=2220
4 MB , 144 MB
HDD 340MB +220
MULTIMEDIA
VIDEO & AUDIO
ZA PROFESIONALNU PRIMENU, EDUKACIJU I ZABAVU
AUDIO KARTICE
OD POPULARNIH DO STUDIJSKIH
(Adlib, Sound Blaster, ProAudio, ..)
VIDEO KARTICE
(Video Blaster, Media Vision, .)
PC TO VIDEO konverteri
FRAME GRABBER
GEN LOCK
LAN
MREŽE
OD JEDNOSTAVNIH KANCELARIJSKIH LAN-ova
DO KOMPLEKSNIH GLOBALNIH SISTEMA (WAN)
Novell Lite, Lantastic
Windows for Workgroups
Windows NT, LAN Manager
Novell Netware 3.12 / 4.01
Unix TCP/IP FTP Software
JUPAK (X 25) Getewsye & Routers
E-mail X 400 sistemi BBS-ovi
PREDUZEĆE ZA RAZVOJ, PROIZVODNJU I PRODAJU INFORMACIONIH SISTEMA I RAČUNARSKE OPREME D.D.
IMTEL computers D.D. NOVI BEOGRAD 11070 BULEVAR LENJINA 165b
Tel.: 011 135-420, 134-516, 135-602, 142-484, 131-749 Tel/Fax: 011 142-164

ДЕДА МРАЗ ЗА ВАШЕ ШТАМПАЧЕ



Пуно тонера и мало сервиса у новој 1994. жели Вам

ПЕРИ ХАРД®
ИНЖЕЊЕРИНГ
Ивана Милутиновића 24
11000 Београд
011/436 019

"Digitalova Alpha AXP arhitektura je vodeći primer kompjutera sledeće generacije i sigurno će ubrzati razvoj prethodno nepoznatih aplikacija "

David Vellems, senior vice president, International Data Corp (IDC)

"U istoriji značajnih proizvoda izbačenih u ovoj industriji bilo je mnogo grubih grešaka Digital se očigledno pripremio za Alpha AXP Verujem da će Digital učiniti jednako dobro sa Alphom AXP kao što je IBM sa AS/400 Snaga Alphe AXP leži u Digitalovoj privrženosti svojim kupcima "

Michael Erbschloe, vice president, Computer Economics, Inc

"Performanse DEC 7000 AXP sistema proteže se od brzog do neverovatno brzog Jedna aplikacija je izvršena tako brzo da to nismo stigli da registrujemo, već smo pomislili da se uopšte ne izvršava "

Joel Richon, computer scientist, Space Telescope Institute, Baltimore, Maryland

"Skala performansi ovih sistema počinje tamo gde današnji 32-bitni RISC sistemi otkazuju Alpha sistemi će omogućiti razvitak softvera za stvaranje aplikacija sposobnih za rešavanje poslovnih i tehničkih problema prethodno smatranih neršivim "

Bob Supnik, vice president, Alpha programme

"Naš signal ide vrlo lako , praktično bez modifikacija Mi imamo vrlo složen set fajlova koje konstrimo postavljene u kalkulacijama, to sve radi bez bilo kakvih zastoja Po pitanju pristupnosti ne bih tražio ništa bolje "

Dr Andrew Holder, president, Semichem

DOBRO DOŠLI U BUDUĆNOST

Procesor koji nas vodi u 21 vek

AlphaAXP

Prezentacija Alpha AXP sistema

1994

Computer Beograd
PROJEKTOVANJE INFORMACIONIH SISTEMA SERVIS I PRODAJA
I SUTJESKA 19a tel 011/711-916
11210 BEOGRAD fax 011/711-478

MUKE STARIH EGIPĆANA

Iako je zvučala prilično naivno, „Egipatska Pitalica" nije ni malo jednostavan problem – nije teško doći do rešanja, ali kada se dođe do pitanja njegova optimalnosti...

Podsetimo se, pre svega, postavke. Neki istoričari matematike smatraju da su stari Egipćani, u cilju lakšeg računanja, predstavljali prave razlomke koje mi danas pišemo u obliku P/Q (pri čemu je P < Q) kao sumu recipročnih vrednosti prirodnih brojeva. Broj 123/456 bi se, na primer, mogao predstaviti kao zbir 1/4+1/51+1/7752. U spisu poznatom pod imenom *Rhind Papyrus* (starost je procenjena na oko 4000 godina) pominju se brojevi u ovom obliku, premda nije precizirano metod kojim bi se neki razlomak doveo u to stanje.

Trebalo je „pomoći" starim Egipćanima, odnosno napisati program koji će bilo koji pravi razlomak pretvoriti u sumu recipročnih vrednosti prirodnih brojeva. Rešenja smo rangirali prema broju sabiraka – što manje sabiraka, bolje je rešenje. Zahtevali smo, prirodno, potpuno tačno predstavljenje – ne dopušta se nikakvo zaokruglјivanje, tj. zbir mora biti **identički** jednak razlomku. Napisani program trebalo je testirati na razlomcima 4609/7320, 1932/32045 i 2377/5413.

Gotovo da i nema potreba dokazivati da se svaki pravi razlomak P/Q može napisati u traženom obliku – ako ništa drugo, moglo bi se to uraditi u formi P sabiraka 1/Q. Problem je dakle, svesti broj sabiraka na minimum za šta, kako se pokazuje, nije lako naći opšti recept. Naši čitaoci su problemu pristupali na razna načina, ali se kao „najprofitabilnija" pokazala ideja da se nađa broj 1/X koji je tek nešto manji od traženog razlomka i da se razlika, P/Q-1/X rekurzivno izlaže istom postupku dok se jednom ne dođe do tačnog predstavljanja. Naravno, za primenu takvog metoda potreban je i matematički dokaz da algoritam u konačnom broju rekurzija dovodi do rešenja.

Pažljivija ispitivanja ovoga metoda pokazaće da rešenja, premda tačna, nisu uvek zadovoljavajuća u smislu broja sabiraka. Razlomak 13/40 bi, na primer, bio predstavljen kao 1/4+1/14+1/280, a u stvari može da se predstavi i kao 1/5+1/8. Prvi sabirak dakle, ne mora da bude baš „najbliži" datom razlomku, već njegov imenilac eventualno treba učiniti nešto većim. Koliko većim? U datom primeru bi moglo da se ide do 1/14, ali je sigurno dovoljno ići do „pola puta" između 1/4 i 1/14. Tako dobijeno rešenje će sigurno kao sabirke imati brojeve oblika 1/X čiji imenioci ne stu

Pošto se formuliše algoritam, treba razmotriti pitanja brzine računara i numeričkog opsega – pojaviće se ogromni imenioci pa je teško održati potpunu preciznost. Čak i sama kontrola rešenja zahtevala je da koristimo neki program koji obezbeđuju „beskrajno" tačnu aritmetiku – opredelili smo se za stari *MuMath* dok su čitaoci koji se redovnije bave numeričkom matematikom sebi obično priuštili paket *Mathematica 386* i to DOS verziju, jer se istomeni paket za *Windows* pokazuje znatno sporijim. Što se računara tiče, brzina nikada nije previše – najbolje bi bilo imati pri ruci neki *DEC Alpha* all, u nedostatku njega, može se „preživeti" i na PC-ju.

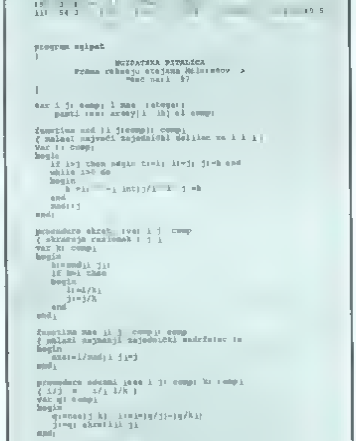
Dobili smo razne programe i rešenja različitog kvaliteta – prvi i drugi razlomak standardno su predstavljani kao zbir četiri sabirka a poslednji kao zbir 7 brojeva. Poneko je postizao i bolja rezultate predstavljajući drugi razlomak kao zbir tri broja, a treći kao zbir 5 ili 6 elemenata. Samo je pet čitalaca „Računara" uspelo optimalno da predstavi sva tri razlomaka, među njima smo izvukli i prve dva nagrađena. Tako će prva novčana nagrada (cena 10 primeraka „Računara") pripasti **Aleksandri Martinović** iz

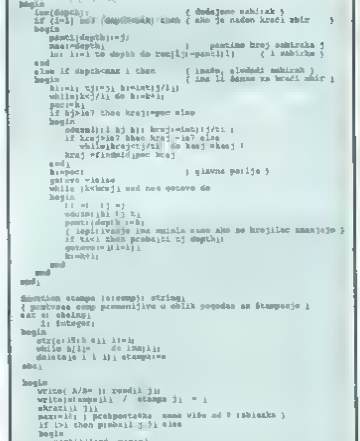
BOŽIĆNI STROJ

Posle jedne od svojih uspešnih godina Napoleon je odlučio da proslavi Božić sa svojim najistaknutijim vojnicima i dodeli im medalje za hrabrost. Svi prisutni su se poredali u stroj prema činovima. Napoleon je najpre prošao duž čitave vrste (počevši od oficira sa najnižim činom) i svakome dodelio po jednu medalju. Zatim se vratio na početak stroja, došao do drugog čoveka sa medaljom i dodelio mu još jednu medalju, a zatim je na isti način nagradio svakog trećeg vojnika u ostatku stroja. Zatim se vratio na početak, preskočio drugog čoveka sa dve medalje i dodelio mu treću medalju baš kao i svakom trećem čoveku u ostatku stroja koji ima dve medalje. Ponovo je došao na početak, preskočio drugog čoveka sa tri medalje a zatim njemu kao i svakom trećem vojniku sa tri medalje u ostatku stroja, dao četvrtu medalju. Postupak je ponovio još nekoliko puta.

Tokom svečanog ručka ispostavilo se da je oficir sa najvišim činom dobio i najviše medalja – čak 10; niko drugi nije dobio toliko. Koliko je bilo ljudi u stroju?

Slika 1





Programsko rešenje „Egipatske pitalice"

KAKO DOSTAVITI REŠENJE

Rešenja Pitalica zajedno sa **obrazloženim** programima koje ste napisali pošaljite na uobičajenu adresu „Računari" (za Dejanove pitalice) Bulevar vojvode Mišića 17, Beograd tako da prispiju **pre** 15 januara 1994. Sva pisma sa korektnim rešenjima konkurišu za novčana nagrade u iznosu cene 10 7 odnosno 5 primerka „Računara" dok će kuponi (ili njihove fotokopije) na koje je upisan identifikacioni broj učestvovati u godišnjem takmičenju rešavača Pitalica. Identifikacioni broj dobijate tako što u prve tri kućice upišete poslednje tri cifre nečijeg broja telefona, u sledeće dve godinu vašeg rođenja i na kraj dodate dve cifre po izboru. Obratite pažnju da sva rešenja koje šaljete u toku godine nose isti identifikacioni broj. Osim običnom rešenja ove Pitalice možete da pošaljete i elektronskom poštom posredstvom Sezame – konsite menije REDAKCIJA, PITALICE, REŠENJE ili iz komandnog moda pošaljite privatnu poruku konsoliku pitalice (*mail send pitalice*)

Novog Sada a druga nagrada (cena 7 primerka „Računara") **Desimiru Pavloviću** iz Beograda. Treću nagradu smo izvukli iz koverta u kome se našlo svih 26 tačnih rešenja (bilo je i tridesetak pogrešnih) – sreća je poslužila **Lidiju Rančić** iz Niša. Pohvale su zaslužili *Filip Crnogorac, Miloš Hrkić, Stojan Miloradović Vladimir Nešić* i *Saša Zeman*. Na slici je program *Stojana Miloradovića* kao i rezultati koji predstavljaju optimalno rešenje zadatka.

Ostaje još da zamolimo našu pobednicu da nam dostavi adresu na koju bismo poslali nagradu (u kovertu nije bilo kupona, a na kovertu nije bilo adrese) i da preporučimo nešto literature onima koje je ovaj problem zainteresovao. Odličan pogled sa istorijske strane naći ćete u članku *J. R. Newman „The Rhind Papyrus"* koji je objavljen 1956. godine u časopisu *The World of Mathematics*, strane 170-178, izdanje *Simon & Schuster*. *R. J. Gillings* je 1962. godine objavio tekst *"Mathematics in the Time of the Pharaohs"*, u kome ćete, između ostalog, saznati kako su Egipćani konstili brojeve oblika 1/X da olakšaju računanje. *M. Dunton* i *R. E. Grimm* su 1966. godine u časopisu *The Fibonacci Quarterly* (v4n4, strane 339-354) objavili tekst *"Fibonacci on Egyptian Fractions"* tu se čitavoj stvari pristupa na mnogo egzaktniji način. Najnovija pronađena referenca na ovu temu je tekst *"Reciprocal Egyptian Notation"* koji su *John Waite* i *Bob F. Wheeler* objavili decembra 1983. godina u časopisu *PPC Journal* (v10n10p30). Rešenje ovog problema, uzgred budi rečano, ima i praktični značaj, koji se naročito ogleda kod projektovanje raznih mreža od pomika.

98: U streju je bilo _____ ljudi

Ime i prezime _____

Adresa _____

Mesto _____

Kompjuter _____ Vreme _____

Identifikacioni broj |_|_|_| |_|_| |_|_| |_|_|
3 broja tel. god. rod. po izboru

NAJNOVIJE TEHNIKE U AUTOCAD-u

Robert M. Thomas

Informacije kod izdavača
BIPIF M.P. za poslovne informacije i izdavačku delatnost
Beograd 27. marta 39/I
tel. 3227-891 fax 3227 321

Knjiga najvećeg svetskog izdavača kompjuterske literature SYBEX-a, sada i na srpskom jeziku

Pomoću nje naučićete da

- kreirate sopstvene tipove linija, šrafura i fontova
- kako da pristupite AutoCAD ovoj osnovnoj bazi podataka radi brže i fleksibilnije kontrole crteža
- modifikujete AutoCAD ov mani

Produbljeno uvođenje u AutoLISP programiranje uz mnoštvo detaljno objašnjenih primera, omogućiće i onima koji ne posjaduju predznanja iz programiranja, pisanje složenih aplikacija!

I sve to je jedinstveno za sve verzije AutoCAD-a 10, 11, 12, ...!

Kupite nekoliko programer godina rada - programe u izvornom Clipper kodu

Knjigovodstveni programi u izvršnom ili izvornom kodu *Finansijsko, Robno, Materijalno, Plate, Trgovačke, Fiskalne, Free Shop, Kumate, Blagajna, Kalkulacije*... Detaljna uputstva za korišćenje, razumljiv i komentarisan izvorni kod. Obezbjeđena telefonska, fax i modemska podrška. Nabavkom stičete pravo na nove verzije programa.

F&D LAB, Podgorica
tel: (081) 37-683; 43-426;
tel/fax: 33-044

HEWLETT PACKARD

LASERSKI ŠTAMPAČI
HP LASERJET 4L/4ML
HP LASERJET 4P/4MP
HP LASERJET 4/4M

MEMO PROŠIRENJA
HP IIp/IIIp/III 1/4Mb
HP 4 4Mb
HP 4 8Mb

POSTSCRIPT
HP IIp/IIp+PLUS
HP IIP/III
QMS HP III 600 dpi
ADOBE HP 4

MICRO 29 NOVEMBRA 71

HEWLETT PACKARD

TONERI
HP IIp/IIIp
HP II/III
HP 4L/4P/4

INK-JET ŠTAMPAČI
HP DESKJET PORTABLE
HP DESKJET PORTABLE COLOR
HP DESKJET 500
HP DESKJET 500C

KERTRIDŽI
HP DESKJET
HP PAINTJET

MICRO TEL.011 343-294 TEL/FAX 343-940

HEWLETT PACKARD

SKENERI
HP SCANJET IIp A4 mono
HP SCANJET IIc A4 kolor

PLOTERI
HP 7475 A3
HP 7550 A3
HP DRAFTPRO A1-A0
HP DRAFTMASTER A0
HP DESIGNJET mono/kolor

RAPIDOGRAF TUŠ
FLOMASTERI
HEMIJSKE OLOVKE

MICRO 29 NOVEMBRA 71

RAČUNARI

VESA LOCAL BUS
'486 SX-25
'486 DX 33
'486 DX 50
'486 DX2 66

VESA LB KONTROLERI
IDE
IDE CACHE
SCSI

GRAFIČKI KONTROLERI
CIRRUS LOGIC
DIAMOND/STEALTH/VIPER

MICRO TEL.011 343-294 TEL/FAX 343-940

EPSON

ŠTAMPAČI
LX-400 A4 9p YU
LX 100 A4 9p YU
FX-1000 A3 9p YU
FX-1050 A3 9p YU
FX-1170 A3 9p
LQ-100 A4 24p YU
LQ-570+ A4 24p YU
LQ-870 A4 24p YU
LQ-1070 A3 24p YU
LQ-1170 A3 24p YU
LQ-2550 A3 24p YU
DFX-5000 A3 9p YU
DFX-8000 A3 9p YU

MICRO 29 NOVEMBRA 71

PRVI GRAĐANIN SVETA

U prošlim „Računarima" upoznali smo globalnu mrežu *Internet* i njanu domaću implamantaciju, *Beointernat*. Opisali emo I neke važne ueluge: elektronsku poštu, prenos detoteka, javna baza podataka... Ovoge meseca videćemo kako izgleda (za mnoga) najintarasantnije usluga umrežanih kompjuterakih aistema – javna konferancija.

Ljudi koje povezuje neki sejednički posao ili hobi oduvek su želeli da međusobno razmenjuju ideje i iskustva da postavljaju pitanja i daju odgovore. U tom cilju osnovana su i osnivaju se razna udruženja i klubovi ali su svi oni u suštini lokalnog karaktera – teško je zamisliti da će neki klas iz Beograda redovno ići na sastanak „Kluba ribolovaca" u Niš e kamoli u Panzu ili Njujorku. Računarske komunikacije su svemu tome dale globalnu dimenziju – ne mičući se iz svoje sobe možete da diskutujete se istomišljenicima iz čitavog sveta, po ceni koja nekto prevazilazi (ili ne prevazilazi) cenu lokalnog telefonskog poziva.

Ovakva mogućnost primene računara uočana je još u samim početcima njihovog umrežavanja pa su se pojavile razna rešenja, većina je počivala na amaterskom radu zasnovanom na entuzijazmu. Obzirom da je „budžet" takvih projekata obično bio približno jednak nuli najpre su nalažena rešenja kojima su se postojeći kanali (naročito privatne elektronska pošta) i postojeći programi mogli upotrebiti se konferancije. Docnije su se pojavili i mnogi komercijalni i sharewara paketi koji omogućavaju diskusiju više konanika a osnovani su i komercijalni konferencijski sistemi. Neki od starih mehanizama i danas lepo funkcionišu pa bar jeden od njih zaslužuje da ga ovde upoznamo – diskusiona lista.

DISKUSIONE LISTE

Koncepcija diskusiona lista vesma je jednostavna. svaki diskutant šalje prilog elektronskom poštom na adresu moderatora liste a onda moderator te priloge sakupiju i takode elektronskom poštom šalje svim „pretplatnicima" – princip mogio bi se reći na koma funkcioniše i većina „papirnatih" časopisa uz dodatak da se nikome ništa ne plaća. Obairom da kroz lista prolazi ogroman broj poruka, bilo bi jako teško obezbediti čoveka („moderatora") koji bi zaista primio i distribuirao poštu. Umesto „živog" moderatora posao obavlja softver pri čemu je većina usluga potpuno automatizovana – postoji adresa na koju se šalju prilozi i druga adresa na koju se šalju zahtevi tehničke prirode (npr. želja da se „pretplatite" na neku listu ili da odjuta nete od njenog praćenja). Ni te zahteve na obrađuje čovek – treba ih stati u precizno određanom obliku koji će program prepoznati i procesirati. Eventualno postoji i treća adresa za koje se krije čovek od krvi i mesa i na koju se obraćaju oni koji imaju problema se nekom od automatskih usluga.

Čitava ova tehnologija naročito je razvijena na *Bitnet*-u, akademskoj mreži o kojoj smo detaljno pisali u „Računarima" 91 i 92 – specijalni program koji obezbeđuju sve što je potrebno za distribuciju lista zove se *listserv*. Postoji ogroman broj lista koje pokrivaju najrazličitije oblasti – od računarske tehnike preko svih grana nauke do hobija i viceva. Same liste svih lista (svaki list pripada tek po jedan red teksta) tranutno se prostire na preko 200 kilobajta – dobićete je ako na adresu *listserv@bitnic bitnet* pošaljete poruku u čijoj će prvoj (i jedinoj) linji pisati *list global* ovo naravno na treba da radite iz Jugoslavije jer je elektronska pošta pod sankcijama pa se treba opterećivati napotrebnim dugim porukama – tih 200 K je već preneseno i može se preuzeti se Sezama. Na slici 1 dato je nekoliko zanimljivih izvoda iz te liste.

Kada odlučite da se pretplatite na neku listu na adresu odgovarajućeg *listserv*-a šaljeta poruku od jednog reda u kojoj piše *subscribe ime liste ime i prezime* gde je *ime_liste* precizan naziv lista na koju se pretpla ćujete a *ime i prezime* vaše „civilno" ime i prezime koje *listserv* na analizira već samo upisuje u svoju internu evidenciju. Adresa na koju se lista ubuduće šalje dekodu je se iz sistemskog dela vaše poruke (zaglavlja o kojima smo u prošlim „Računarima" govorili jesu složena ali se iz njih može izvaditi mnogo vrednih informacija) pa je na treba navodit u okviru *subscribe* naredbe. Sledećeg dana elektronskom poštom ćete vam stići potvrda de je zahtev primljan i uspešno obrađen uz a ventualno neki tekst opšte prirode koji opisuje listu na koju sta se pretplatili i pravila ponašanje „u njoj". Ubuduće će vam redovno (sve dok ne iste adresu na pošaljete komandu *unsubcribe*) stizati tekstovi koje drugi upućuju u izabranu listu ako ih neko vreme nema, to ne mora da znači da nešto nije u redu naprosto ima li sta u kojome je saobraćaj jako slab. Možda da sami pošaljete neku poruku koja će podstaći diskusiju?

Što se slanja priloga tiče, postoje dve vrste lista – moderisane i nemodensane. Kod nemoderisanih prilog se naprosto šalje na neku adresu (u uvodnom tekstu kojim se lista predstavlja biće navedena i ta adresa obavezno je prepišite) i odatle biva automatski distribuiran svim pretplatnicima (pa i vama). Kod moderisanih lista prilog se takode šalje na posebnu adresu ali odatle nešta biti distribuiran dok ga moderator na pregleda i na odobri. Ovakva lista obično pruža znatno kvalitetniju diskusiju – moderator će svakako izbaciti poruke koje nose malo ili nimalo informacija (nrp „kilometarski" citat i lepod toga trjava „sasvim se slažem") „sasači u konsnu" razna vređanja, umanjiti broj skretanja se tema itd tpak ovakve liste su znatno „eporije", pošto moderatoru vođenje liste obično pred stavlja sporedni posao pa se dešava da denima na stigne da odobri nove priloge, dok prilori stignu mnogi od njih već i zaborave o čemu su uopšte razgovarali! Zato ćete dosta zadovoljstva naći i u nemodensanim listama, mada su za njihovo praćenje ponekad potrebni neiito jači nervi.

Diskusione liste nisu „specijalitet" *Bitnet*-a odnosno EARN-a i na *Internetu* je zastupljen ovaj vid komunikacije. Na primer poruka koju biste poslali na adresu *sun-managers@secs.nwu edu* automatski bi bila razaslata svim pretplatnicima ova liste administratorima raznih *Sun*-ovih radnih stanica. Administrativni poslovi (npr. pretplate na liste) obično se obavljaju slanjem poruke na adresu koja je jednaka imenu liste ali se sufiksom *-request* na primer *sun-managers-request@secs.nwu edu*. Treba, međutim, znati da se ovakvi zahtevi na *Internetu* obično ne rešavaju automatski već ih rešava moderator više puta nam se dešavalo da nas moderator denima ne uključi u listu i da moramo da željemo po nekoliko puta dok na prvučemo njegovu pažnju. To je međutim, mali problem za autora ovoga teksta je mnogo već bio kada se „neopraano" pret plalo na neku listu koja je pokazalo se donosila ogroman broj poruka koje se jednostavno nisu mogla pratiti. Gotovo 15 dana sam molio moderatora da mi izbaci se liste, a svale dan je stizalo po par stotina kilobajta mani nepotrebnog teksta. Pazite, dakle na šta se pretplaćujete i budite spremni na iznenađenja!

Listu svih lista na *Internetu* možete da preuzmete FTP-om (u prošlim „Računarima" videli smo kako se FTP koristi za prenos datoteka) se računara *ftp.nisc.sri com*, anenimni pristup je dopušten, a lista se nalazi u direktorijumu *netinfo* i zove *Internet-groups*. Ako vam više odgovara elektronska pošta, možete se obratiti na adresu *spaf@cs purdue edu* – listu održava *Gene Spafford*.

Iako ćete pregledajući ovu „listu svih lista" naći mnoga zanmljiva naslove diskusione liste nisu glavni vid konferisanja na *Internetu*. Većina poruka prenosi se *Usenet* grupama pri čemu se neke grupe naknadno „reimituju" kao *Internet* pa čak i kao *Bitnet* liste što je od velike pomoći konanicima čij računan obezbeđuju samo prijem elektronske pošte a ne i druge usluge svetske mreže.

KRATKA ISTORIJA Useneta

Usenet se u formi koja je na prvi pogled znatno različita od današnje pojavio sredinom 1979. godine. Dva studenta *Duke* univerziteta u Severnoj Karolini (USA), Tom Traskot (*Truecot*) i Džim Elis (*Jim Ellis*) razmišljali su o povezivanju fakultetskih računara u cilju lakše razmena informacija o upravo izašlom *Unix*-u V7. Na osnovu njihovih ideja Stiv Belovin (*Steve Bellovin*) student susednog univerziteta napisao je prvu verziju programa (nazvanog *news*) i instalirao je na tri univerzitetska računara. Ta mreža koja se sastojala od tri čvora opisana je januara 1980. na *Usenix* konferenciji u Bulderu (*Boulder CA*).

Čitav sistem je startovao jako dobro i izazvao veliko interesovanje ali i saobraćaj koji je daleko prevazilazio mogućnosti prvobitno napisanog softvera. Zato je tokom 1981. godine *Mark Horton* student sa Berklija uz pomoć srednjoškolca Meta Glikmana (*Matt Glickman*) napisao novu verziju kojoj je dao oznaku *news_B*. posle jednogodišnjeg beta testiranja ova verzija je ušla u širu upotrebu da bi docnije bila više puta doradivana. Poslednje verzije čiji je autor Mark Horton nosi oznaku 2.10.1.

Tokom 1984. godine rad na unapređenju ovoga softvera preuzeo je Rik Adams (*Rick Adams*) iz Centra za sezmičke studije u Kaliforniji. Dodao je mogućnost kreiranja modensanih grupa da bi krajem 1986. napravio veliku reviziju softvera i obezbeđio dosledniie imenovanje grupa olakšao prenos podataka uz njihovu moguću kompresiju dodao nove kontrolne poruke. Tu je nagde i bio kraj „B" revizije *news*-a – poslednja verzija nosi oznaku 2.11.19.

Tekuća verzija *Usenet* softvera, koju označavamo slovom C nastala je na Univerzitetu u Torontu – razvili su je Geof Kolier i Henri Spenser (*Geoff Collyer and Henry Spencer*). Motiv za njen nastanak je isti kao i motiv za nastanak svih prethodnih – *Usenet* je toliko porastao da su se morale preradivati čak i rutine najnižeg nivoa da bi se racionalno podržao takav internzet saobraćaja, izbeglo dupliranje poruka olakšala selekcija informacija i unapredila pouzdanost čitavog sistema prenosa članaka. Nova verzija je počela da se primenjuje od jeseni 1987. godine i u upotrebi je do danas naravno uz određene modifikacije koja se neprekidno vrše. Detalji se mogu naći u radu *News Need Not Be Slow* koji je publikovan u zbirci saopštenje sa *Winter 1987 Usenix Technical Conference*. Obzirom da se *Usenet* softver nalazi u javnom vlasništvu, i međuvramenu je portiran na mnoga platforme, među kojima važno mesto zauzimaju IBM PC kompatibilni računari.

ŠTA JE Usenet?

Evo jednog sasvim prirodnog pitanja na koja uopšte nije lako odgovoriti – među onima koji prvi put pristupaju *Usenet*-u čuće se mnogo pogrešnih interpretacija njegova definicije i svrhe i pokazaće se da su neke od tih interpretacija spravo glavni krivci za povremeni „rat porukama" u nekim grupama.

Usenet je kratko rečeno ogromna grupa računara koji razmenjuju članke. Članci se razvrstavaju po standardizovanim naslovima kojima su podeljeni u tekstvene grupe (u originalu se koristi izraz *newsgroups*

ali se često čuju i termini „konferencije" ,oblasti', „okrugli stolovi" SIG-ovi (SIG = *Special Interest Group*) ,forumi" itd) Organizacije kojima pripadaju svi ti računari su veoma raznolike – tu su razni univerziteti naučna institucije i škola državna agencije poslovne organizacije amaterske grupe i mnogi drugi Ni jedna od tih organizacija (uključujući tu i državu koja, nagde ima monopol na komunikacione linije koje se koriste se razmenu informacija) nema kontrolu nad čitavim *Usenet*-om – svaki administrator kontroliše svoj čvor i nema posebnog uticaja na administratore drugih čvorova Ovlašćenja administratora počiču od vlasnika datog sistema – sve dok je vlasnik zadovoljan radom svog administratora, nema nikakve „spoljne sile" koje bi ga smanila ili naterala da svoj „atom" *Usenet*-a drugačije održava

Osim što ne postoji nikakvo centralno telo koje bi rukovodilo *Usenet*-om i sama topologija je krajnje „amorfna" Postoji neka pređutna hijerarhija čvorova u kojoj se smatra da su neki veći računari (koji komuniciraju brzim linijama preko *Internet*-a) „gore' a da su milioni malih (često i personalnih) računara koji jednom dnevno na UUCP principima razmene konferencije „dole" Osim toga, činjenica da će neki od većih računara prenositi neku konferenciju znači da će sv računari koji „vise" na njemu biti u prilici da je (ako žele) prenose čima tej veći sistem ostvaruje neku „kontrolu" nad manjim sistemima koji se njim komuniciraju Ipak, svaki od tih sistema će ako želi da prati i neke konferencije koje obilžnji računar na prenosi lako stupiti u kontakt se nekim drugim *Internet* sistemom i prenositi poruke (i) sa njega Kako su brzine komunikacija raste (optička vlakna itd) veći sistemi su prenosili sve više i više konferencija tako da danas vlasnik svakog računara može precizno izabrati šta ga tačno zanima i onda to pribavi uz sasvim male troškove

Prenos poruka podrazumeva i određena troškove pa se mogu naći organizacije koja se novac obezbeduju prestup *Internetu* i *Usenetu* (npr UUNET) Ipak za ogromnu većinu korsnika pristup je besplatan – stvar je u tome što su se mnoge organizacije priključile na *Internet* zato što su im trebale razne usluge te globalna mreža (posebno elektronska pošta) To priključenje se tako što smo videli u prošlim „Računarima" svodi na nabavku potrebne opreme i sekup komunikacionog kanala (npr telefonske linije) do „susednog" *Internet* sistema. Kada je linije već zakupljena, nama razlike da li će preko nje danas preći megabajt ili sto megabajta podataka Pošto je intenzitet saobraćaja koji generiše sama ta organizacija često sasvim mali i ograničen na radno vreme nema prepreka da se u periodima „praznog hoda" prenose konferencije koje će možda zanimati neke zaposlene a ako i posle toga linija ostaje neiskorišćena, i ostale konferencije *Usenet* uopšte funkcionišu na osnovu dobra volje vlasnika i administratora više miliona računara – većina njih je spremna da rezerviše neki deić svojih resursa da bi se omogućilo globalno širenje informacija.

Prethodne konstatacije se odnose uglavnom na Sjedinjene Države i Kanadu u kojima je, kao što smo videli, *Usenet* i nastao Sračom, *Usenet* nije samo američka *mreža* – veliki broj računara u Evropi, Australiji i Japanu takođe prenosi *Usenet* diskusije brojne izuzetno uspešne konferencije nastale su upravo u Evropi i sada se prenose po čitavom svetu Osnovni jezik na kome se komunicira je engleski mada su tommane i neke grupe lokalnog tipa u kojima se razgovara na drugim jezicima – jedna od njih je i nama dobro poznate grupa soc.culture.yu u kojoj već par godina žestoko polemišu ljudi iz raznih republčika bivše Jugoslavije Ipak, pitanje finansiranje se u nekim evropskim organizacijama znatno „formalnije" rešavalo a ponegde je i država pokušala (ili uspela) da uvede monopol na prenos ovih informacija. Poslednjih godina se, zahvaljujući sva jeftinijem prenosu podataka „preko okeana" javlja sve više organizacija koje se povezuju direktno se američkim *Internet* sistemima tako da polako nestaje pričino „amorfni" evropski *Usenet* Šta se domaće situacije tiče sankcije onemogućavaju da se dođe od potpunog rešenja činjeni su pokušaji da se uz pričine JUPAK troškove prenose određene *Usenet* grupe na domaće sisteme ali su neki korsnici u njima mogli samo pasivno da učestvuju čitajući poruke – mogućnost odgovaranja na postoji Naravno korisnici koji imaju pristup nekim stranim sistemima i koji su spremni da

Nekoliko zanimljivih izvoda iz pregleda diskusionih lista

plaćaju komunikaciona (JUPAK) troškove njihovog zvanja, u prilici su da, na način opisan u tekstu „Sa VAX-om po svetu" iz „Računara 65" u punoj meri učestvuju u *Usenet* konferencijama

Letos je u okviru *Beoinfonets* instaliran *Usenet* softver i krairano vile diskusionih grupa iz najrazličitijih oblasti – ona će se prenositi na veći domaće sisteme ali neće biti u vazi se globalnim *Usenet*-om (o tim domaćim grupama i njihovom praćenju biće reči u sledećim „Računarima") Zato je jezik na koma se diskutuje srpski, po uključaju sankcija verovatno će se stać uklonje da aktivno učestvujemo u svim diskusijama koje se u svetu vode

Sa čisto tehničke strane posloje dva metoda za transport poruka: UUCP i NNTP UUCP je bazian na standardnim modemima i telefonskim linijama – poruke koje su korisnici napisali ,čekaju" u nekom baferu dok njihov sistem (npr jednom dnevno) na pozove susedni računar i na prosledi mu ih Sistem istovremeno preuzima nove poruke u grupama koje njegovi korsnici prate a naki drugi računari decnije mogu sa njega preuzimati neke od tih grupa.

NNTP je primeren sistemima koji su stalno vezani sa *Internetom* Nova poruka se šalje suseđnom sistemu gotovo u momentu kada je napisana, a „u realnom vremenu" pristižu i poruke se drugih sistema Zahvaljujući dobro napisanim transportnim programima svaki sistem može preuzimati konferencije se više desetina drugih sistema – neki prenose jednu konferenciju, neke drugi drugu i tome slično. treba samo u konfiguracionom fajlu naveati šta se uzima odakle.

i UUCP i NNTP razlikuju poruke po identifikacionom broju – svaki sistem mora da obezbeđi da svake poruke ima jedinstven identifikacioni broj Identifikacioni brojevi su obično dugi i krpični i koranku na znače praktično ništa za *Usenet* softver su, međutim, jako važni jer onemogućavaju dupliranje poruke (ista poruka, usled amorfne strukture mreže ponekad stigne se dva strana) Osim identifikacionog broja i *email* adrese autora, zaglavlje *Usenet* poruka sadrži i mnoge druge informacije – pogledajte sliku 2 na kojoj su prikazana dve *Usenet* poruke sa raznih sistema i iz raznih grupa.

ŠTA Usenet NIJE?

Naša predstava o tome šta je *Usenet* neće biti potpuna ako ne razumemo i šta *Usenet* nije Zabluda o njemu nekoamo nije malo i mnoge od njih su često tehničke prirode ni jedna od tvrdnji „*Usenet* je *Internet*" „*Usenet* je *Unix* mreža" „*Usenet* je ASCII mreža", „*Usenet* je UUCP mreža" „*Usenet* je akademska mreža" „*Usenet* je komercijalna mreža" nije tačna! A ipak se mnoga od njih tako često čuju, možda i zato što u svakoj ima po zrnce istne

Usenet zaista nije *Internet* niti UUCP jer *Internet* i UUCP mreže obezbeđuju mnoge druge usluge osim konferencija a *Usenet* se čita i na mnogim računarima koji nisu vezani ni u UUCP ni u *Internet* mrežu Ipak glavnina komunikacije ide preko računara vezanih u *Internet* i prenosi se od „krajnjih korsnika" preko UUCP sistema koji takoda komuniciraju sa *Internet* računarima. *Usenet* dakle nije ni *Internet* ni UUCP ali je u uskim vezama i sa jednim i sa drugim Slično važi i za odnose *Usenet*-a i *Unix*-a videli smo da je *Usenet* softver portiran na razne platforme ali je sva počelo na *Unix* sistemima, a i nove verzija se pojavljuju sličnim putevima. Velika većina *Usenet* poruka je zasta, pisana u ASCII-ju ali nas ponekad iznenadi i neki EBCDIC tekst

Sve pomenute zablude nisu posebno bitne na praktvode neko negativno dejstvo (osim kada poneko u poruci koristi PC set znakova iznad 128 pa se to na raznim VAX-ovima vidi kao niz „kuka i kvaka") Neke druge zablude međutim proizvoda nepijatne posledice, naročito po onoga ko deluje na osnovu tog pogrešnog uverenja Na primer pitanje vlasništva *Usenet* nije univerzitetska niređa iako su mnogi veliki čvorovi koji prenose konferencije upravo univerzitetske institucije najzad, *Usenet* je i nastao na američkim univerzitetima Međutim *Usenet* nije ni komercijalna organizacija, iako je većina uključenih računara vla snićno nekih komercijalnih organizacija Zato je pitanje oglašavanja na njemu jako složeno pošto se lako može desiti da se kroz računara jedne organizacije prenose reklame za konkurentske proizvode Mnogi su spremni da to „trpe" ali do odsedanje mere – dok je reklama u formu informacija i dok se pređesto ne ponavlja sve je obično u redu ali se ponekoma naprosto desi da njegov sistem „ispadne" iz *Usenet*-a Za posiovne informacija (pa i za reklame) koriste se *biz* gru pe u kojima vladaju precizinje pravila i koja se prenose samo na neke od sistema.

Druga zabluda, odgovorna se mnoga diskusije na samo na *Usenet*-u nego i na raznim BBS-ovima je po sledica (pogrešnog) verovanja da na *Usenet* u vlada demokratija sloboda štampe ili nešto slično tome. Stvar postaje još gora kada se, recimo u našim uslovima, pojave (obično mladi) korisnici kojima nije baš jesno ni šta je „demokratije" ili „sloboda štampe" već imaju o tome neka svoja krajnje idealizovana uverenja. Jedino pravo koje postoji na *Usenet*-u je pravo vlasnika nekog sistema da na svom računaru drži ono što želi i da ne drži ono što ne želi Iz toga prava proizlazi sve ostalo – ako vlasnik nekog sistema ne želi da preko njega prolazi neka grupa čiji se sadržaj protivi njegovim uverenjima, tu je kraj priče o toj grupi na tom sistemu! Isto tako nema nikakve svrhe da korsnici nekog sistema „glasaju" o tome da li će se prenositi ova ili ona grupe – sve i da je velika većina za prenošenje, posledniju reč ima vlasnik Naravno na sistemima koji rade na komercijalnim osnovama vlasnik obično obrača pažnju na želje korsnika jer od toga na kraju krajeva, zavisi uspešnost njegovog sistema, ali definisanje poslovne strategije sistema svakako nije stvar njegovih korsnika Delava sa da vlasnici nekih sistema naprosto neće da prenose poruka nekog korsnika drugog sistema (ili čak bilo kakve poruka se tog drugog sistema) iz ovog ili onog razloga – tom korisniku (ili tom sistemu) ostaje da traži druge kanale da učestvuje u *Usenet* diskusijama ili ako mu „svi" u okolini „olkažu ljubav" digne ruke od čitavog posla.

Što se „slobode govora" tiče u *Usenet* diskusija ma zbilja šeta pročišćavanja i svaka – ograničenje golovo i da nema. Ipak, ako neko na svom sistemu ništa da vidi vašu poruku on će je se njega ukloniti i neće je dalje prenositi na vredi vam da se pozivate na neka kva svoja „prava na i repliku", „slobodu štampe' i tome slično Najzad u Americi postoji izreka koja kaže (slobodno prevedeno) „sloboda štampa je pravo onih koji poseduju štampu"

I na kraju pitanje autorskih prava. Kod nas je što se piratvanja programa tiče situacija takva kakve je i mnogima se čini sasvim „prirodnim" da u konferenciji traže da prsnkne neki program plaju kako se može nelegalno registrovati *Telemate* i tome slično. Takvi korisnici se začude već i kada im na nekom domaćem s-

sistemu (Seram, FDN ETF) poruka bude obrisana – nijedan ozbiljniji domaći BBS na tolerše poruke kojima se vređaju autorska prava stranih i domaćih programera. No najgore što se nekome može desiti je da mu pristup bude uskraćen i da mu „propadne" pretplata koju je eventualno platio. U stranim konferencijama posledice mogu da budu mnogo ozbiljnije i obuhvataju novčane i zatvorske kazne. Teško je istina, pomisliti da će neko biti gonjen preko *Interpola* zato što je pirat tevao program ali ako se jednom nađe u Americi možda će saznati da je „neko" sve to negde evidentirao i da ga presuda čeka. Čak i bez toga, visnici „usputnih" sistema preko kojih poruka prelazi ne žele da oni budu izloženi eventualnoj tužbi što su prenosili poruku kojom se nečija prava vređaju – najlakše stvar za njih je da zabrane prenos poruka određenog korisnika (ili određenog čvora) tej korisnik će se onda naći u situaciji da na može da pošalje ni ostale poruke kojima traži neku samim legalnu uslugu. Ulaktko *Usenet* na smete koristiti kao pomoć pri nelegalnoj razaci komercijalnog softvera – načete ga tim putem nabavit ć možete imati i vrlo na prijatnih neugodnosti

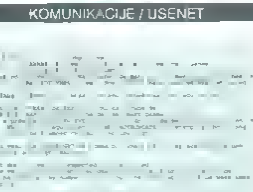
IMENOVANJE GRUPA

Imena *Usenet* grupe pomalo podsećaju na FQDN imena *Internet* računara o kojima smo govorili prošlog meseca – nekoliko reči razdvojenih tačkama, na primer *rec.humor.funny*. Ime je u osnovi hijerarhijsko – prva reč najgrublje opisuje na šta se grupe odnosi druga određuje tematiku a treća je ona sama grupe, ponegde se sreću i imena koja se sastoje od više reči. U osnovi postoje šest oblasti – *comp* označava grupe koje se bave računarstvom (softver hardver algoritmi kompjuterske nauke itd), *sci* grupe se bave raznim (uglavnom prirodnim) naukama, *soc* se odnosi na društvo, kulturu politiku i slično teme tek su ruma debelina grupe u kojima je primarni cilj „očiranje pera" (treba li reći „očirenje tastatura"?), a na prenos naših naročito korisnih informacija, *rec* su grupe koje se bave rekreacijom hobijima i sličnim temama dok su *misc* grupe koje se na mogu uklopiti ni u jednu od pomenutih ka tegorija. Postoje osim toga i grupe *news* koje se bave samim *Usenet*-om u okviru kojih se diskutuje o poboljšanju novih grupa, *Usenet* softveru i drugim sličnim temama. Smatra se da svaki korisnik *Usenet*-a treba da prati bar grupe *news.groups* i *news.announce.newgroups* kako bi bio u prilici da se izjasni o eventualnom pokretanju novih grupa ili glasanju neaktivnih ili uopšte da bi bio informisan o najvažnijem što se dešava na *Usenet*-u

U pravi *Usenet* konferencije **nikako** na treba da šaljete teet poruke kojima biste ustanovili znate li da koriste softver koji ste nabavili da li je vaš sistem dobro povezan i tome slično. Tome su namenjene razne test grupe (najpoznatije je *misc.test*) u kojima i nema „normalne" diskusije već se sve svodi na testiranje. Neke od tih grupe uključuju čak i program koji će automatski „odgovoriti" na vašu poruku uz citiranje delo njenog sadržaja kako biste videli da je sve dobro prošlo i znali kako će ubuduće izgledati odgovori na vaše poruke.

Osim pomenutih, pojavile su se i (u početku) „nezvanične" (mada je na *Usenet*-u i inače malo toga „zvaničnog") grupe koje se bave oblastima koja neki na žele da prenose. To su pre svega, grupe čija imena počinju se *alt* – u njima vlada „anarhija" a i teme su prilično „alternativne", na primer seks droga, perverzije, „mesni" vicevi. Očnovane su i *biz* grupe koje se bave raznim poslovnim informacijama. Ni jedne ni druge mnogi sistemi na žele da prenose ipak, raznovrsnost i difuzna struktura *Internet*-a omogućće onima koje te teme zanimaju da nađu način se njihovo praćenje

Čak i u nekim „konvencionalnim" grupama uobičajeno je da poruke koje po mišljenju autora, mogu da povrede nečije stavove (npr rasistički vicevi) budu na neki način „šifrovane" Uobičajeni metod zove se *rot13* – svako slovo se u takvim porukama pomera se 13 mesta, naviše' tako da A postaje N ll postaje O itd. Eventualni čitalac te poruke mora pokrenuti mali program koji je vraća u „normalan" oblik (mada smo videli bar jednog korisnika koji uma da čita ove poruke i bez ikakve transformacije. praksa čuda pravi) – ako se i pođa toga oseća uvređenim, na može reći da mu je iko za to kriv jer je očevidno sam želeo da pročita tu poru-



Dve *Usenet* poruke sa raznih sistema i iz raznih grupa.

ku! Sličan metod koristi se i za „skrivanje" delo diskusije u nekim konferencijama ako bi neka računica mogla nekome da pokvari slobodni „užitak" (npr diskusuje se o nekom kriminalističkom romanu, a u tom delu računice pomenuto je ko je ubica)

Većina *Usenet* grupe nije moderisana – sve što napišete „ide u svet" U nekim grupama se, međutim, prilozi šalju moderatoru koji će ih razmotriti i eventualno uključiti u razmenu – moderator koji ima vremena ponekad se odlučuje da na pošalje sve primljene poruke (koje se često dobrim delom poklapaju po tematici itd) već da u grupu pošalje *digest* se značajnim navodima iz tih poruka. Čitaoci su tako u prilici da pročitaju samo ono što je (po mišljenju moderatora) bitno čime štede svoje vreme. ali su zato uskraćeni za raznovrsnost informacija. Svako će dakle, pratiti one grupe koje ga zanimaju i koje se vode na način koji mu se dopada

POKRETANJE NOVIH GRUPA

Šta da rade oni koji su pregledali spisak postojećih *Usenet* grupe i u njemu nisu našli onu koja se bavi njihovim poslom li njihovim hobijem? Takva situacija nja naročito verovatna ali je saavim moguća – najraći i *Usenet* treba da raste dodavanjem novih grupa!

Otuzrom da *Usenet* kao što smo videli nije nikakva centralizovana organizacija (bolje reći nje nikakva organizacija) nema centralnog tela kome se treba obratiti u cilju kreiranja nove grupe. Možete dakle na svom računaru kreirati novu grupu i eventualno ubediti par prijatelja da je prenose na svoje računare Takve grupa, međutim ima jako male šanse da se proširi – vlasnici većina sistema neće ni znati da ona postoji i samim tim neće je prenositi ni svojim korisnicima niti drugim računarima koji su se njima povezani Grupa će dakle, ostati neka vrsta lokalnog foruma za dogovore nekoliko prijatelja i u tome nema ništa loše ! *Usenet* softver se često koristi baš za takve stvari Pe lpak, ako želite da osnujete grupu koja bi se globalno prenosila i u čijem bi radu učestvovalo mnogo ljudi moraćete da sledite izvesnu uobičajenu proceduru

Kada se rađi o kreiranju nove grupe, prva osobina koju treba da pokažete je strpljenje – ako hoćete da vaša grupe uspe morate na tome pažljivo raditi od samog početka A taj početak je praćenje grupe *news.announce.newgroups* i *news.groups* u toku šest ili više mesaci – se to vreme videćete kako se grupe kreiraju kako teče diskusija, šta je važno a šta nije Zlatno pravilo kaže da onaj ko je manje od 6 meseci pratio ove grupe a ipak misli da je spreman da predloži kreiranje nove treba da čeka na 6 mesaci nego godinu dana! Kada osetite da ste spremni kreirate „u napad" – treba pokrenuti globalnu diskusiju o eventualnom pokretanju grupe načinu njenog rada i moderatorima. Prvi korak je takozvani RFD (*Request For Discussion* poziv na diskusiju) koji se šalje u grupu *news.announce.newgroups* a takođe i u druge grupe koje se bave srodnim oblastima. Grupe *news.announce.newgroups* je moderisana, što znači da će moderator najpre proveriti da li je predlog usklađen sa „dobrim običajima" *Usenet*-a, baš zbog upoznavanja tih običaja više mesaci ste pratili ovu grupu Diskusija o vašem predlogu ubuduće će se vodih u konferenciji *news.groups* – tokom te diskusije (koje bi trebala da traje bar mesec dana) iskristalisaće se precizno ime grupe seključni da li će ona buda moderisana ili nemoderisani i ko bi bili eventualni moderatori

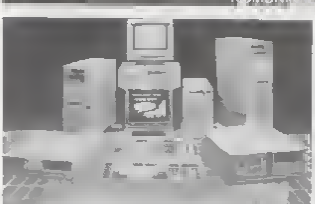
Posle 30 (ili više) dana diskusije predlagač može u *news.announce.newgroups* (opet posredstvom moderatora) poslati poziv za glasanje zvani CFV (*Call For Votes*) U okviru poziva mora se postaviti precizno pitanje tipa „Da li ste za formiranje grupe koja se zove tako-i-tako bave se tim-i-tim i čiji je moderator taj-i-taj i mora se objasniti kako se glasa – obično slanjem elektronske pošte na neku adresu pri čemu se u temi poruka (*Subject*) piše YES ili NO Ako se očekuje veći priliv glasova pogodno je otvoriti ove adrese, pri čemu će na jednu stezati glasovi **za** a na drugu glasovi **protiv** I stom prilikom se objavljuje i koliko će glasanje trajati obično 21 ili 31 dan Tokom tog perioda nema diskusije o grupi i nema objavljivanja preliminarnih rezultata ali se eventualno može ponavljati poziv za glasanje Kada je glasanje završeno rezultati se objavljuju u *news announce newsgroups* pri tome se objavljuje i spisak *email* adresa onih koji su glasali i uz svaku od njih da li je korisnik glasao za ill protiv na taj način svako može da proveri da li je njegov glas korektno evidentiran i sebran Sledi period od oko pet dana u kome se mogu iznati primedbe na ispravnost glasanja. Ako nema ozbiljnih primadbi moderator *news announces.newgroups* će proveriti rezultate – ako je preko dve trećine glasača bilo za I ako je glasova se bar za 100 više nego glasove protiv šalje se obaveštenja administratorima svih većih sistema da je grupe kreirana i da je mogu preneseti Prilična je verovatnoća da će se ako je procedura poštovana, većina njih zaista i odlučiti da prenosi grupu a vremenom će i vlasnici manjih sistema uraditi nešto slično Grupe tako počnje da živi.

Ako glasanje nije uspelo kroz šest mesaci se može pokušati još jednom sa istim ili sličnim predlogom Ako ni posle nekoliko ovakvih pokušaja kreiranje grupe „na prođe' ostaje vam samo da odustanete od posla ili da kreirate lokalnu grupu na svom računaru

LEPO PONAŠANJE U MREŽI

U ogromnom broju poruka koje svakodnevno putuju *Usenetom* nije lako izdvojiti one koji su (za nas) interesantne od onih drugih baš to izdvajanje predstavlje najveći problem za one koji ne mogu 24 sata dnevno ekspti i čitati poruke De bi se problem umanjio od svih koji pišu poruke se očekuje da budu obzirni prema drugima i da sledе nešto što se takom zadnjih desetak godina formirako pod imenom *Netiquette* ili slobodnije prevedeno, „Bonton *Usenet*-a" Pravila tog bontona nisu ni mnogo čvrsta ni bez izuzetaka ipak, od vas kao početnika u tom svetu očekuje se da ih se pridržavate koliko možete

Prvo i ubedljivo najvažnije pravilo je na **šalite besmislene poruke** Mnogi početnici pogotovu mladi misle da treba da odgovore baš na svaku poruku koju pročitaju Odgovor se obično svodi na veliki otst iz poruke i onda jedan rad komentara koji je po mišljenju onoga ko ga piše jako duhovit' za većina čitalaca komenter je na prosto glup Nije lako biti duhovit. e ni vrhunskom komediografu na bi uspelo da svakoge dana buda duhovit po 100 puta jedno što tim odgovorima postižete jeste da nervirate ljudi koji su trošli pare da ju vašu poruku prenesu i vreme da je pogledaju Posle nekoliko „duhovitih" poruka zaradićete od mnogih „crvenih karton" – instruisaće svoj program za praćenje *Usenet*-a da ignoriše vaše poruke; kada vam jednom



nešto bude trebalo ili kada kažete nešto zaista pametno mnogi neće pročitati vašu poruku i ostaćete bez željenog odgovora. Trudite se ukratko da pišete poruku samo ako imate šta da kažete i ako ima izgleda da će to što kažete biti korisno potencijalnim čitaocima.

Pri izboru grupa u koju ćete slati novu poruku treba da budete pažljivi – neko kome je stalo da se na njegovo pitanje što pre odgovori poslaće istu (dugu!) poruku u nekoliko srodnih grupa i tako samo iznervirati one koji sve te grupe prate (ako su grupe srodne velika je verovatnoća de je i auditorijum sličan). Umesto toga u jednu od grupa koje najbolje odgovaraju onoma o čemu pišete možete da pošaljete samu poruku, a u srodne grupe možda samo *pointer* (reč-dva) u kome ćete ukratko o čemu ste pisali i reći u kojoj grupi je same glavna poruka

Pre nego što postavite pitanje vredi da razmislite o tome da li je to pitanje možda već stotinama puta postavljano U tom cilju se za većinu grupa, formira i održava takozvana FAQ lista često postavljanih pitanja i odgovora na njih Svakako proverite da li vaše pitanje spada u takva. Čak i ako ne spada naše škodilo da pročitate FAQ i tako saznate odgovore na neka pitanja koja ćete možda kasnije poželeti da postavite

Sledeće pravilo je da na korisničke konferencije za privatnu poštu – ako poruka koju ste poslali iz nekog razloga ne stiže ili vam ona kojoj ste pisali ne odgovara, **najglupije** što možete da uradite je da krenete da šaljete tu poruku u konferenciju koje ona, po vašem mišljenju može prati (umesto toga napišite privatnu poruku *Postmaster*-u njenog računara i pitajte da li ime problema se vašom porukom) Smatra se čisto toga, vrlo ružnim odgovarati na nečiju privatnu poruku u konferenciji – privatna poruka treba da ostane privatna sem ako se i pošiljalac složio da je proslediti u konferenciju Obmuta logika ne važi – naprotiv dobro je u nekim situacijama odgovarati na konferencijska poruke privatnom poštom zapravo uvek treba dobro razmisliti da li je bolje da odgovor pošaljete u konferenciju ili da predate na privatnu poštu Ako je neko postavio neko trivijalno pitanje u konferenciji ča možda stići više desetina ili stotina manje-više identičnih odgovora Zar mađu njima mora da bude i vaš? Isto tako ako se se nečim slažete nema potreba da tim slaganjem opterećujete konferenciju – možete ga izložiti i u privatnoj poruci Ako se na slažete od vas se očekuje da obrazložite zašto – poruka se velikim citatom i komentarom „na slažem se" čitaocu ne znači baš ništa.

Pri pisanju odgovora obično se citira tekst na koji se odgovara – na taj način čitalac lakše prati diskusiju, jer je poruku na koju odgovarate možda pročitao još pre nekoliko dana pa je i zaboravio o čemu se radilo **Nikako** na citirajte čitali tekst, samo neko njegov deo (par redova) koji najbolje opisuju ono što je sagovornik napisao i da na to odgovorite Očekuje se da vašeg teksta bude više od citiranog to je, najzad, vaša poruka Takođe smatra se vrlo ružnim „secrati" nečiju poruku reč po reč vaditi računice iz konteksta i naročito manjih nešto u onome što je sagovornik rekao

Citati se obično obeležavaju znakom veće (>) ili dvotačkom ( ) Neki korisnici se „dosete" de uokvire citat nacrtaju „sanku" itd ali takve ideje neko nailaze na dosar prijem jer nepotrebno produžavaju poruku i otežavaju rad raznih *off-line reader*-a koji vrlo ne mogu da odrede šta je citat a šta tekst.

Svaka poruka koju pošaljete će prirodno počinjati vašim imenom i *email* adresom se koje ste je poslali. Ipak to ime nije uvek jednako vašem „pravom" imenu pa mnogi osećaju potrebu da poruku i „potpišu" Većina programa za slanje poruka omogućava da u svom osnovnom katalogu kreirate datoteku koja će se recimo zvati *signature* i čiji će sadržaj automatski biti dopisan ispod svake poruke koju pošaljete Nevolje se tim potpisom je što je tako prepoznati – neki tu napišu čitavu svoju biografiju pa je još dopune mislima velikih ljudi koje cene a kada im to nije dosta onda i nacrtaju neku ASCII vinjetu (nako se čak nađe pametan da u ASCII setu „crta" svoj „svojeručni" potpis) Sve to je možda zanimljivo kada se prvi put vidi ali zamislite nekoga ko danas čita deset vaših poruka a svaka ima taj neseretni potpis od pola ekrana! Potrudite se da potpis na bude duži od dve ili tri reda, a ako ikako možete da „preživite" bez njega bolje i da ga ne stavljate!

I se kraj dve važne stvari koje se povremeno javljaju i totalno „ubijaju" svaku konstruktivnu diskusiju u pojedinim grupama. Prva od njih je pisanje tipa „Da li je bolje da kupim Atari ST ili Amigu" postavljeno u neko grupi koja okuplje vlasnike jednog ili drugog računara. Sasvim je moguće da vas zanima odgovor na to pitanje, ali verujte nam njegovo postavljanje u konferenciji definitivno nije način da do tog odgovora dođete! Ono što će se desiti jeste „religiski rat" u kome će se, često bez posebnih argumenata, svađati vlasnici jednog i drugog kompjutera, pokušavajući drugima (a naročito sebi) da dokažu da su oni izabrali baš onaj pravi Te kvs rasprave je možda prvih par puta zanimljiva za praćenje ali iz nje nećete mnogo toga naučiti niti ćete na osnovu onoga što pročitate doneti neku odluku Najbolje je da pogledate jedan i drugi računar i da sami odlučite a eko vas neka pojedinost zanima, odgovor ćete svakako dobiti u odgovarajućoj *Usenet* konferenciji

Drugo pitanje se svodi na ukazivanje na jezičke greške – na Sezamu smo više puta primetili da u nekoj diskusiji korisnik koji ostane totalno bez argumenata u prilog teze koju je zastupao nađe neku jezičku grešku u poruci „protivnika" citira je i proprati komentarom tipa „Ša ti znaš o tome-i-tome kad na znaš pravilno ni da pišeš!" Nevolja je što onda „druga strana" uzme tu poruku pa u njoj nađe nekakve jezičke greške (nije ih teško naći gotovo u svakoj poruci) pa onda krene njih da komentariše i obrazloži u toj temi zapodene jezičke rasprave koja nema nikakve veze sa onim o čemu se prethodno diskutovalo i koja ne zanima veliku većinu onih koji tu temu prate Na *Usenet*-u je ovaj problem još akutniji jer su diskusije na nekom od jezika gde se na greši samo u gramatici nego i u spelovanju (obično engleski) pa se greške još lakše nađu Ovakvo skretanje diskusije smatre se za ozbiljan *faux pas* Ako ste se ista zainteresovani za raspravu o jeziku vodite je u odgovarajućoj konferenciji a ude poruke koristite možda samo kao primer a ne kao način da se omalovažavate autor ili njegovi stavovi

NAJŠIRI FORUM

Šta reći na kraju? *Usenet* grupe su najširi postojeći forum za računarsko konferisanje i omogućavaju razmenu mišljenja na najraličitije teme se korisnicima računara iz svih krajeva sveta. To je velika prednost *Usenet*-a ali možda paradoksalno, i njegova velika mana. Pokazuje se da je „nivo šume" ti broj za nas neinteresantnih poruka koje moramo da pročitamo pre nego što nađemo na neku koja nas zaista zanima dosta visok, a činjenica da većina korisnika pretupa konferencijama beoplatno na naki način snižava nivo diskusije Na nekim profesionalnim sistemima kao što su *CompuServe* ili IIX broj korisnika je mnogo manji, ali su relativno visokim pretplatama (i drugim troškovima) ti konenci „filtrirani" na profesionalce iz neke oblasti. pa je nivo diskusije viši e „nivo šuma" često znatno prihvatljiviji Pa ipak se nekoga ko se uključuje u svet globalnih komunikacija *Usenet* je mesto od koga treba početi videćete dokle se sve može stići i do kakvog se obilja vrednih informacija može doći a sve to bez previsokih troškova!

Sledećeg meseca videćemo kako rade *Usenet* konferencije na *Beointernet*-u i kako im se pristupa se Sezama FON-a i ETF-a.

Nastavak sa strane 29

UDARNI NOVITETI

Glavni novitet MS DOS-a 6 2 se, kao što sa i očekivalo, odnose na sigurnost podataka. DBLSPACE se pokazao veoma pouzdanim, mnogo pouzdanijim nego što smo ako ćemo iskreno, očekivali Pe ipak, pronađene je jedna čudna situacija u kojoj su DBLSPACE (program koji „udvostručava" kapacitet diska) i SMARTDRV (program za keširanje) dolazili u koliziju, u tom slučaju program koji se izvršava „pogađe" u zonu memonje koju SMARTDRV samo što nije prepisao na disk. Što dovodi do oštećenja podataka Taj bag je ispravljen, ali su autori DOS-a, plašeći se očito neka slične kolizije, ugradili mehanizme koji treba da je unapred spreče Ti mehanizmi dobili su ime *doubleguard* i svode se na proveru integriteta podataka pre njihovog upise na disk Provera se zasniva na složenom *checksum*-u i prame rečime Microsoft-a, „ni ze mrvicu" ne usporava rad sa diskom Usporenje „za mrvicu" je uvek teško izmeriti, ali neme sumnje da novi DBLSPACE zauzime primetno manje RAM-a od prethodnog – svega 33 kilobajta Može se, kao i ranije, premestiti u visoku zonu memorije dodavanjem parametra /MOVE u CONFIG SYS Oni koji zaključe da DBLSPACE, i pored svih oplsanih unapređenja, nije dovoljno siguran, sada mogu da ge uklone bez gubitka podataka (pretpostavlje se naravno, da je kapacitet nekompnmovanog diska dovoljan za sve postojeće podatke) – opcije *uncompress* će prebaciti podatke ne „običan" disk, postepeno emanjujući kompnmovani, sve dok podaci ne budu bezbedno uklonjeni iz zone odgovornosti DBLSPACE-a. Nepostojanje ove opcije je bile jedna od naših glavnih zamerki MS DOS-u 6 0

Obezbeđenje sistema primećuje se i ne mnogim drugim mestime HIMEM.SYS će, na primer automatski testirati čitavu memoriju u potrazi za eventualnim nesprevnim čipovima – Microsoft-u je svakako dosadilo de konsnici koji, stalno „viseći" neku zastarelu verziju *Lotus*-a ili *Tetris*-a, ne znaju da im je računar nesprevan pa okrivljuju DOS 6 za probleme koje izaneade imaju se diskom „Pojačanje" je i komande SCANDISK, koja pregleda površinu diska u potrazi za ukrštenim i izgubljenim klasterime i fizičkim oštećenjime magnetnog medija. Uz DOS 6 2 više nećete morati da nabavljate slične alatke je paketa *Norton Utilities* ili *PC Tools De Luxe*

Nova sintaksa i semantika nekih komandi takođe je usmerena preme obezbeđenju podataka – SMARTDRV na primer, u startu neće keširati upis ne disk već će to aktivirati tek ne zahtev korisnika. Komanda MOVE, COPY i XCOPY prepravljena su tako de traže potvrdu pre nego što prebrišu postojeći fajl – već se godinama pitamo zašto nema tog tako potrebnog kontrolnog pitanje! Uvedena je još jedna stvar koju već duže vreme očekujemo od Microsofta. DISKCOPY će koristeći proširenu memoriju ili proetor na disku, sada prekopirati svaku disketu (pa i one od 2 88 megabajta) u jednom prolazu Dodatne memorije dobro dođe i nekim drugim uslužnim programima kao što je DEFRAG – uspeli smo da komprimujemo čak i jedan disk od 120 megabajta na kome se nalazile datoteka od preko 110 megabajta, prethodna verzija DEFRAG-a je vrlo brzo „pucala", doduše bez oštećenja podataka ali i bez obavljenog posla.

IZUZETNO DOBRA STVAR!

Za razliku od eksperimentalnih jezika koji se mogu videti po laboratorijama, većina „stvarnih" programskih jezika je vaoma slabo opremljena za detekciju i ispravljanja grašaka. Ukoliko u vašoj apilkaciji doda do graška pri čitanju podataka sa diska, taško da imata ikakvu altarnativu sem da zaustavita program. Evo radikalno boljeg rašanja!

Dejan Jelović

Aks je irski seter s greškom Ne radi se o tome da njemu stvarno nešto fali – ja ga zovem tako da bih nervirao njegovu gazdaricu Njegov nedostatak je u tome što je njegova dlaka pomalo bleda. Umesto bogate crveno-mahagoni boje koja krasi većinu njegovih erodnika, Aks je u najboljem slučaju, svetlo braon

Kad kompenzacija za ovu genetsku grešku Aks je obdaren derom ne kome bi mu pozavideo I najbolji tažni propovednik Sposoban je da vas netere da mu date sve što imate samo jednim svojim pogledom Kada sedna i upan svoje tužne oklce prema vama sadržaj vašeg frižidera će se garantovano naći u njegovom stomaku Njegova najveća specijalnost su deca, koju začkuje u prolasku pored njegova kuće Smoki, čips, užina ill bilo šta drugo brzo postaju „putarina" na putu do obližnje osnovne škole

Aksova ljubav prema hrani je i osnov za njegovo vaspitanje Za perče pite spremen je ne samo de sedne ill legne već i de dubi na trepavicame ili de izvrši bilo koju drugu nemoguću komandu koju smislite A čuvena komande „k nozi" se uz nešto hrane u džepu lako pretvara u „pripj sa uz mene koliko možeš" Postoji samo jedne stvar koju Aks voli više od hrane, a to su mačke Kade vidi mačku, njegov mozak trenutno doživ „Divide overflow", svest mu se pomrači, i u trenutku na vidi ništa osim nje Možete de imets i pečenog vola, sve to gubi značaj u odnosu na mačku – dolazi do totalnog raspada sistema. Sledećeg momenta on se daje u paničnu jurnjavu za njom I na ostaje vam ništa do de ga progonite nemarskim prostranstvima.

JEDAN DAN U ŽIVOTU PROGRAMERA

- Slog je zaključan
- Konsnik je izvadio disketu u toku prebacivanja podataka ne nju radi backupa
- Uneti podaci izazivaju „Divide overflow"
- **scanf** traži de unese broj, e on sivonja upisuje rečima.
- Indeks je neažuran – verovatno je laključio kompjuter dok je program radio
- Nema više mesta na disku
- Nema memonje – **malloc** mi vraća NULL.
- Ako ovde izvršim **salt (1)**, ostaće mi nezatvorene datoteke Mislim, trebe prvo izprazniti bafere

IZUZECI

Postoji odredena klasa problema u programu koji se nazvaju „nepredvidena situacije" Pod time se na misli na direktne greške koje i zazivaju da program instant padne, već na neke lude situacije kojih ima toliko de ih jednostavno nje realno predvideti sve Manje-više svaka bibliotečka funkcija vraća neki status, ali se on, osim na strateškim mestima, jednostavno ne proverava, jer to na bi bilo realno

Mehanizmi koje većina programskih jezika nudi za hvatanje ovakvih ekscesnih situacija su veoma rudimentarni Svode se mahom ne pozivanje neke predefinisane funkcije (error handler) ili na zaustavljenje programa. Interpretrarni jezici nude i nekakvu ON ERROR GOTO konstrukciju, ali ni ona ne funkcionišе baš savršeno. Prava stvar u ovom slučaju au *exceptions* – izuzeci

Izuzeci predstavljaju način de se za svaki blok komandi koji nešto radi napravi blok zadužen za hvatanje grešaka. Još bolje, hvatanje grešaka može de bude koncentrično: blok za hvatanje grešaka se napravi u glavnom programu, a onda on pozive razne funkcije u kojima se nalaze njihovi blokovi za hvatanje grešaka. E sad, ukoliko do greške dode, nju prvo prima blok iz funkcije, pa onda on može na osnovu greške de nešto uradi, ili de prosledi grešku dalje kroz hijerarhiju. To omogućuje de se strukturnan pristup – od opšteg ka specifičnom – primeni i na greške Može se, recimo, u glavnom programu napraviti blok koji kada do greške dode otvori prozor sa porukom o grešci i onde zaustavi program, e u pojedinačnim funkcijama blokovi koji nake greške umeju de „isprave" (ali ai ga oprevio, svaka ti časti), a one koje ne mogu prosleđe glavnom programu

Problem se izuzecima jeste što nema mnogo kompajlera koji bi ih podržavali Na PC-ju to za sada radi samo *Borland C++* za *Win32* (alfe verzija), što baš i nije pohvalno Strustrup je izuzetke definisao još pre par godina u čuvenom ARM-u ali ih je, zbog komplikovane implemetacije, slabo koje firma ubacila u avoje kompajlere Ovde ću zato izneti način de se neki podskup izuzetaka implementira u C-u e ni C++ programeri neće ostati kratkih rukava. Ono što je zanimljivo jeste da i bilo koji drugi jezik koji podržava neku vrstu **#define** direktive može bez problema da implementira ova kvazi izuzetke Oni se na kraju krajeva, svode na nekoliko makroa i funkcija. No, pre nego što se pozabavimo implementacijom hajde de prvo vidimo kako to izuzeci konkretno funkcionišu.

PROBAJ ME, PROBAJ...

Program koji podržave izuzetke zasnovan je na takozvanim **try** i **catch** blokovima Posle službene reči **try** ide blok naredbi u kojime može doći do greške Odmah iza **try** bloke nalazi se **catch** blok, zadužen za hvatanje greške koje su nastupile, **try** blok se izvršava normalno kao i svaki drugi deo programa, dok se **catch** blok izvršava samo ako do greške dode Pogledajte na listingu 1 kako to izgleda.

Dakle, program se najnormalnije izvršava unutar **try** bloka, e **catch** blok biva pozvan sa kodom greške ukoliko do greške dođe Iz primera se vidi i naredba **throw** – one prosleđuje izvršavanje na neki **catch** blok u funkciji koja je pozvale našu funkciju Ili u neki pre toga Ukoliko se izda **throw** a de nigda ranije u programu nema **catch** bloka, program se zaustavija

Da u terminu „greška" ne bi bilo nikakve magije, recimo i to da se greške generišu isto funkcijom **throw** Recimo, greška DIVIDE OVERFLOW se generiše negde u programu jed nostavnim

```
throw (DIVIDE OVERFLOW);
```

Ono što je specifično za izuzetke jeste de se **throw** može pozvati bilo gđe kako u samom programu tako i u nekom interaptu Ne gornjem primeru hvatanje deljenja se nulom, **throw** se poziva u interaptu koji mikroprocesor izdaje svaki put keda do deljenja sa nulom dođe Isto tako se mogu hvatati situacije kada nema dovoljno mamorije (operator **new** u tom slučaju poziva *user-defined* funkciju), pritisne se Ctrl-C ili bilo koja druga situacija koja vam padna na pamet. Moguće greške zapravo sami dodajete Jednostavno, negde u programu kažete

```
#define GRESKA_TA_I_TA 200
```

i onda, tamo gde je potrebno, pozivate **throw**

```
throw (GRESKA_TA_I_TA);
```

Izuzeci sadrže mnogo više nego što sam je ovde napisao – širu sinteksu više **catch** blokove u odnosu ne jedan **try** blok koji se onda pozivaju po tipu parametra (na primeru je parametar uvek **int** ali zapravo ne mora biti tako) i slične poslastice Međutim, pošto mi ovde nemamo načina da sve to implementiramo, ostanimo, za sada, ne ovome što možemo

IMPLEMENTACIJA

Implementacija izuzetaka se svodi na **setjmp/longjmp** funkcije De vas podsetim, funkcije **setjmp** „pamti" sliku sistema – stek gde je program stao, registre, flegove i slično Mi ćemo naše **try** i **catch** makroe napraviti ne osnovu njih (vidi sliku) Dve stvari treba naglasiti

1 Iako onginalni C++ izuzeci dopušteju da se u **catch** blok prosledi bilo šta (int, char *, neki objekat, ...) mi smo u našoj implementaciji ograničeni na jedan tip podataka. Radi portabilnosti uvodimo tip **Exception** koji se definiše pomoću **typedef** konstrukcije Iako je **int** nekako logičan kao izbor tipe promenljiva koja bi nosile kod greške on ima fu manu da se lako može desiti da dve odvojene biblioteke definišu

Listing 1

```
[illegible]

#define [illegible]

int main()
{
        try {
                throw 100;
        }
        catch (int i) {
                cout << "Caught " << i << endl;
        }
        catch (double d) {
                cout << "Caught " << d << endl;
        }
        catch (...) {
                cout << "Caught default" << endl;
        }
        return 0;
}
```

Primer programa za obradu izuzetaka

Listing 2

```
[illegible]

// Maximum number of handlers  try  a and catch[illegible] number
// of objects[illegible] we also[illegible]
#define MAXTRY 20
#define MAX_MAIN_OBJECTS 50

                        DEFAULT EXCEPTIONS

const Exception OUT_OF_MEMORY = "Out of memory";
const Exception ARITHMETIC_ERROR = "Arithmetic error";
const Exception CONTROL_C_PRESSED = "Ctrl-C";

Exception currException;
jmp_buf jmp_bufStack[MAXTRY];
int jmp_bufCounter = 0;[illegible]

void fail(Exception e)
{
        for (Exception *
        if (jmp_bufCounter) {
                while (*declarerBase::topObjectPtr)
                        (*declarerBase::topObjectPtr)->~dclr();
                longjmp(*jmp_bufStack[--jmp_bufCounter]);
        }
        else
                abort();
}

//                  IMPLEMENTATION OF declarerBase

declarerBase* declarerBase::objectsStack[MAX_STACK_OBJECTS] = { NULL };
declarerBase** declarerBase::topObjectPtr = objectsStack;

declarerBase::declarerBase()
{
        *topObjectPtr++ = this;
}

                        DEFAULT EXCEPTION HANDLING

struct NO_DEFAULT_EXCEPTION
{ NO_DEFAULT_EXCEPTION() }

NO_DEFAULT_EXCEPTION throwNever;

void operator new(size_t)
{
        throw OUT_OF_MEMORY;
}

void operator delete(void*)
{
        throw ARITHMETIC_ERROR;
}

main catchDEFAULT()
{
        throw CONTROL_C_PRESSED;
}

static class DefaultExceptions {
        public:
        DefaultExceptions() {
                set_new_handler(outOfMemoryHandler);
                signal(SIGFPE, catchDEFAULT);
                signal(SIGINT, catchDEFAULT);
        }
} defaultExceptions;

#endif
```

Listing 3

```
                Copyright (C) 1993 [illegible] Jelo[illegible]  rights reserved
                Yes can [illegible] use this file in your programs as is or modified
                You can not publish major parts of [illegible] by or them permission

#ifndef __EXCEPT_HPP
#define __EXCEPT_HPP

#include <setjmp.h>

typedef char* Exception

//                          DEFAULT EXCEPTIONS

extern const Exception OUT_OF_MEMORY
extern const Exception ARITHMETIC_ERROR
extern const Exception CONTROL_C_PRESSED

//                          CLASSES

class declarerBase {
        friend void fail(Exception e);

        static declarerBase** objectsStack;
        virtual void dlr() = 0;

        public:
        static declarerBase** topObjectPtr

        declarerBase();
        ~declarerBase() { topObjectPtr--; }
};

template <class T>
class declarer : public declarerBase {
        T* obj;
        public:
        declarer(T* obj) : obj(obj) {}
        void dlr() { delete obj; }
};

#define declare(c) declarer<c> obj__declared__(c)

//                          INTERNAL VARIABLES

extern Exception currException;
extern jmp_buf* jmp_bufStack[];
extern int jmp_bufCounter = 0;

//                          MACROS

#define try {
                jmp_buf errorBuf;
                int exception;
                jmp_bufStack[jmp_bufCounter++] = &errorBuf;
                if ((exception = setjmp(errorBuf)) == 0)
                {

#define catch(E)
                        jmp_bufCounter--;
                        declarerBase::topObjectPtr
                }
                else if (exception == currException)
                {
                        E = currException;
                        if (exception != 0)

void fail(Exception E)
inline void fail() { fail(currException); }
#define throw fail

#endif
```

Try i catch makroi

```
#include <iostream.h>
#include <stdlib.h>
#include <setjmp.h>

class testclass {
        char name;

        public:
        testclass(char n) { name = n; cout << "Entering " << name << "\n"; }
        ~testclass() { cout << "Exiting " << name << "\n"; }
};

void test()
{
        testclass a('a'), b('b'), c('c');

        int i = random(100);
        cout << 100 / i << "\n";
        cout << 100 / 0 << "\n";
}

main()

{
        try {
                randomize();
                for (;;)
                        test();
        }
        catch (Exception e)
                if (e == OUT_OF_MEMORY)          cout << "Out of memory\n";
                else if (e == ARITHMETIC_ERROR)  cout << "Arithmetic error\n";
                else if (e == CONTROL_C_PRESSED) cout << "Ctrl-C\n";
                else throw;
        return 0;
}
```

šu jednu vrednost za dve različite greške. Zato je **char*** mnogo logičniji izbor.

(Ovo treba konsti i kada implementiramo dinamičku tipizaciju – ukoliko objekat treba da vrati informaciju kojoj klasi pripada, mnogo je bolje da vraća neki string, nego da se pravi neka konstrukcija tipa.

```
#define GEOMETRIC_SHAPE 1
#define CUBE 2
#define CIRCLE 3
```

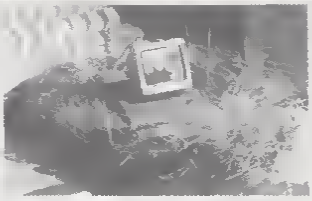
jer se lako može desiti da dve različite biblioteke uzmu broj četri, jedna za, recimo, liniju, a druga za elipsu.)

2. Kada se pomoću **throw** izvrši skok u neki **catch** blok, za sve objekte na steku bi trebao da se pozove destruktor. Ovo je trivijalno za izvođenje ako se ugradi u kompajler, ali je zato apsolutno nemoguće to naknadno uraditi. Zato u ovu implementaciju pomoću šablona i makroa uvodimo direktivu **declare**:

```
String s ("test");
declare (String, s);
```

koja je zadužena za deklaraciju objekata. Sada, kada se izda **throw** automatski bivaju pozvani svi destruktori objekata koji su na steku, sve dok se na naiđe na **catch** blok. Naravno **declare** se može „uneti" u konstruktor, pa se na taj način deklaracija objekata može vršiti automatski, s tim što se onda taj objekat ne može naslediti – tj. može ali nema svrhe.

Naravno, sve bi se ovo moglo izbeći ukoliko bi imali jednu klasu tipa **Object** koju bi sve druge klase nasleđivale, ali to jednostavno nije realna



pretpostavka u C++ programima. (To je način na koji *NIH Class library* implementira izuzetke.)

U ovoj implementaciji su već definisani izuzeci za aritmetičke greške Ctrl-C i nedostatak memorije, ali ih, ukoliko ne želite da opterećuju vaš kod, možete slobodno izbaciti. Ne sliči mate i mali demo koji pokazuje kako izuzeci rade.

Sve u svemu, izuzeci se implementiraju veoma jednostavno – **try/catch/throw** sa mogu bez problema implementirati na bilo kom jeziku u svega tridesetak linija koda, s tim što posebna „zezanja" oko pozivanja destruktora zahtevaju još malo koda; treba definisati osnovnu klasu odgovarajući šablon i makro za deklarisanje objekata na steku.

WIN OR LOSE

S obzirom da su sve firme za koje radim manje-više prešle na 386 ili jače mešine, počeo sam da pišem programe za *Windows*. Da ne bude zabune, još uvek mislim da su *Prozori* očajna implementacija dobre ideje, što ne mora da ide na dušu Majrosoftu, već se može pripisati kompatibilnosti sa DOS-om. Ostaje činjenica da *Windows+DOS* u tri puta više prostora smeštaju upola manje funkcionalnosti neko Mek. Kao što to reče Mič Kapor, „Bill Gates has won. The computer industry is the empire of the dead."

No, kada smo već tu, iskoristimo *Prozore* koliko god možemo. Sa programerske strane, *Prozori* nude DLL-ove koje definitivno treba maksimalno (zlo)upotrebljavati. Ne samo da bi se izbeglo ponavljanje istog koda u više različitih programa, već i da bi se ostvario stari san o korišćenju različitih alatki na jednom projektu. Sada bez problema možete da kritične delove vaše aplikacije uradite u C-u, a korisnički interfejs u nekoj vizuelnoj okolini ala *Visual Basic* ili nešto još bolje. Da ne spominjem koliko se zbog jedinstvenog korisničkog interfejsa sve te različite alatke dobro uklapaju. Ili pošto već imate minimum *Notepad*, nema potrebe da vaša baza podataka pravi svoje module za unos memo polja – o tome može da se brine *Windows*, a korisnik ima tu prednost da memo polja ukucava u svom omiljenom editoru.

Gledajući iz perspektive korisničkog interfejsa, velike je pomoć što je sve već napravljeno za vas – od običnog prozora do najkompliKovanijih dijaloga. Ili, moje poslastica, dodavanje zvuka u programe. Zamislite samo prijatan ženski glas kako vas obaveštava da slog koji hoćete da menjate već neko koristi ili da je štira koju ste unali pogrešna.

„Hello Gentlemen. I am a HAL 9000 computer. I became active in the HAL plant in Urbana, Illinois, on January 12, 1992."

386 SX-40MHz AMD

RAM 2Mb
FD 1.2 Mb
HD 170 Mb
SVGA 512K TRIDENT
14" SVGA MONO MONITOR
MINI TOWER, TASTATURA 101

1400 DEM

386 DX-40MHz AMD

RAM 4Mb, 128Kb CACHE
FD 1.2 Mb
HD 170 Mb
SVGA 512K WIN.AKC.
14" SVGA MONO MONITOR
MINI TOWER, TASTATURA 101

1740 DEM

486 DX-40MHz AMD

RAM 4Mb, 128Kb CACHE
FD 1.2 Mb
HD 215 Mb
SVGA 512K WIN.AKC.
14" SVGA MONO MONITOR
MINI TOWER, TASTATURA 101

2290 DEM

DISKOVI

HARD DISK 170 Mb 18ms	450 DEM
HARD DISK 215 Mb 16ms	500 DEM
HARD DISK 258 Mb 12ms	550 DEM
HARD DISK 340 Mb 12ms	700 DEM
HARD DISK 450 Mb 12ms SCSI	1200 DEM
HARD DISK 540 Mb 9ms IDE III SCSI	1350 DEM
FD 1.2Mb / FD 1.44Mb	120 / 90 DEM

KARTICE

16 bit IDE CONTROLER / 32 bit VL BUS IDE CONTROLER	40 / 130 DEM
32 bit VL BUS IDE CONTROLER 4Mb CACHE	850 DEM
SCSI ISA / EISA CONTROLER 4Mb CACHE	1000 / 1250 DEM
ETHERNET 16 bit NE 2000	130 DEM
FAX MODEM KARTICA MNP5 - HARDVERSKI + V42 bis	180 DEM

SVGA

SVGA 14"MONITOR 1024x768 MONO / COLOR	280 / 650 DEM
SVGA 17"COLOR MONITOR 1280x1024 RAVAN EKRAN LOW RAD.	2150 DEM
TRIDENT 512Kb	100 DEM
CIRRUS 5422 WIN. AKC.1280x1024 512Kb / 1Mb	140 / 180 DEM
32 bit VL BUS CIRRUS 5426 WIN.AKC.1280x1024 1Mb	220 DEM
32 bit VL BUS S3 WIN. AKC. 1280x1024 1Mb	330 DEM

PLOČE

386 SX-40 MHz AMD	160 DEM
386 DX-40 MHz AMD, 128Kb CACHE	280 DEM
486 DX-40 MHz AMD, 128Kb CACHE	780 DEM
VL BUS 486 DX-33 MHz INTEL, 256Kb CACHE	850 DEM
VL BUS 486 DX-50 MHz INTEL, 256Kb CACHE	1150 DEM
VL BUS 486 DX2-50 MHz INTEL, 256Kb CACHE	1200 DEM

PRINTERI

LASER HP IV L 300x300	1000 DEM
LASER HP IV 600x600 / RAM 4Mb	3500 / 450 DEM
A4 EPSON LX 400 / LQ 570	480 / 900 DEM
A3 EPSON FX 1170 / LQ 1070	1150 / 1250 DEM

RAZNO

MEMORIJE SIMM 1Mb / 4Mb	90 / 330 DEM
MATEMATIČKI KOPROCESOR 387DX-40MHz	120 DEM
MINI TOWER 200W / MIDI TOWER 200W	130 / 160 DEM
PROFESIONALNA TASTATURA / ALPS SA KLIKOM	70 / 110 DEM
MIŠ - MICROSOFT KOMPATIBILAN	50 DEM
STREAMER 250 Mb SA TRAKOM	550 DEM

DOPLATA: 14" COLOR MONITOR + MIŠ = 400 DEM

POSEBNE CENE ZA DILERE

VESA 486DX-33MHz INTEL

RAM 4Mb, 256Kb CACHE
FD 1.2 Mb+1.44 Mb
HD 215 Mb
32bit VL BUS CONT.
32bit VL BUS SVGA 1Mb -
- WINDOWS AKCELER. 1280x1024
14" SVGA COLOR MONITOR
MINI TOWER, TASTATURA 101, MIŠ

2990 DEM

VESA 486DX-50MHz INTEL

RAM 4Mb, 256Kb CACHE
FD 1.2 Mb+1.44 Mb
HD 215 Mb
32bit VL BUS CONT.
32bit VL BUS SVGA 1Mb -
- WINDOWS AKCELER. 1280x1024
14" SVGA COLOR MONITOR
MINI TOWER, TASTATURA 101, MIŠ

3290 DEM

VESA 486DX2-66MHz INTEL

RAM 4Mb, 256Kb CACHE
FD 1.2 Mb+1.44 Mb
HD 210 Mb
32bit VL BUS CONT.
32bit VL BUS SVGA 1Mb -
- WINDOWS AKCELER. 1280x1024
14" SVGA COLOR MONITOR
MINI TOWER, TASTATURA 101, MIŠ

3400 DEM

ISPORUKA ODMAH GARANCIJA 12 MESECI

ŽIRO RAČUN: 60815-601-2-64787 RADNO VREME 9h-17h

TEL: 011/332-607
FAX: 011/345-126

PREDUZEĆE
ŠUTLIĆ
&
MIKRO DIZAJN

11000 BEOGRAD
Kosovska 32, I sprat

JANUAR 94.

PROŠIRENI DOS

Prošli put smo se uponali aa postojanjem i istorijatom nastanka DPMI protokola, koji je kroz *Windows* operativni sistem (?) uneo pravu revoluciju u kompjutersku industriju. Međutim, pre nego što krenemo u ovu još uvek nejasnu i slabo korišćenu oblast, upoznaćemo se sa najbližim rođakom DPMI sistema – DOS ekstenderima.

Bojan Petrović

Kada je onomad Ray Duncan čitao poverljivu specifikaciju DPMI protokola (tekst iz prošlog broja) na OS/2 simpozijumu, reče mi mogao da pretpostavi koliko će se svet promeniti zahvaljujući Microsoft ekipi. Ne samo da je na osnovu toga nastao *Windows 3.0* nego se razvile i cela industrija softvera u zaštićenom modu. Međutim u vreme nastanka DPMI protokola na tržištu su već postojali sistemi koji su koristili zaštićeni mod procesora, među kojima su najznačajniji bili DOS ekstenderi. Na nesreću ondašnje DPMI specifikacije, ovo dvoje su se sudarali u mnogim aspektima rada tako da je na kraju došlo do „van pamćenog" dogovora. DPMI je odustao od direktne podrške prekida i funkcija DOS-a, a sa uzvrat dobio podršku celog tima koji je stajao iza tada već razvijenog VCPI. Tako su DOS ekstenderi ostali kao neophodni ustav za rad DOS aplikacija u zaštićenom režimu a DOS je oživeo još jednu mladost. Naravno danas smo već zastrašujuće blizu nekom novom DOS sistemu koji bi „normalno" radio u zaštićenom modu procesora ali o tome kada se desi.

A dok se na desi DOS ekstenderi su jedini način rada sa 32-bitnim aplikacijama pod DOS-om *Windows* sistem nudi svoje rešenja se problem koje ne zahteva DOS ekstenderi. U čemu je dakle smisao ove tehnologije? Pre svega, oslobađanje memorijskih limita na se svim legalan način povećanje snage DOS aplikacija i nov „mali korak ali veliki za čovečanstvo" u softverskoj industriji.

TRADICIONALNI DOS EXTENDER

Čak i pre pojava *Windows 3.0* sistema, na tržištu su se mogli naći mnogi programi koji su svojom veličinom i snagom zbunjivali prosečnog korisnika. Recimo takvi su bili *Mathematica Paradox 386 AutoCAD* i drugi. Kupovinom jednog ovakvog programa zapravo dobijate dva. DOS ekstender licenciran od neke firme koja ga proizvodi (a to obično nije firma čiju aplikaciju kupujete) i samu aplikaciju. Firma koje proizvodi predmetnu aplikaciju od neke druge firme kupuje licencu za redistribuiranje DOS ekstendera (jedan od najpoznatijih proizvođača je Phar Lap) i direktno ga ugrađuje u svoj kod (ličnije, na početak EXE fajla koji vi startujete). Pa startovanju jedne takve „sendvič" aplikacije MS-DOS zapravo učitava DOS ekstender i startuje ga, bez ikakvog polmenja aplikacija koja je smeštena posle njega. Što je u obliku EXE zaglavlja, u kom je opisano postojanje samo samo ovog prvog DOS učitava DOS ekstender u konvencionalnu memoriju u realnom modu odvojivši mu potreban prostor i predaje vlast njemu. On zatim odvaja potrebnu količinu memorije učitava ostatak EXE fajla (što je zapravo aplikacija), izvodi sve potrebne zahvate u sistemu po trebne za rad u zaštićenom modu i konačno straruje aplikaciju. MS-DOS nije u stanju da učitava programe za zaštićeni mod iz prostog razloga što nije sveštan postojanja zaštićenog mode, selektora ili bilo čega u vezi 32 bitnog rada.

Zahvati potrebni se rad u zaštićenom modu mogu biti veoma komplikovani. Pre svega formira se tabela prekida tabela lokalnih i globalnih deskriptora kao i poseban segment namenjen adresiranju kada se skoči u zaštićeni mod. Ovde se obično formiraju i selektori (pokazivači) za mapiranje video bafera PSP-a progra ma koji se obrađuje BIOS podataka i drugih potrebnih struktura. Konačno, DOS ekstender prebacuje CPU u zaštićeni mod i preklapa kontrolu aplikaciji.

Kada jednom aplikacija krene se radom, DOS ek stender postaje veza između aplikacije u zaštićenom modu i DOS sistema u realnom modu. I to potpunom kontrolom svih prekida. Zapravo DOS ekstender se može shvatiti kao rezidentni program koji što se DOS-e tiče raziduje u memoriji čiji je vlasnik aplikacija. I samim tim nestaje se njenim prestankom rada. DOS ekstender mora presresti svaki prekid bilo da ga on ek sternog tipa (sat) internog (deljenje nulom) ili softver ski. U većini slučajeva kada nastupi prekid u zaštićenom modu, DOS ekstender prekida prebacuje CPU u realni mod „dobacuje" poruku o prekidu origi nalnoj rutini za obradu prekida, i čeka kraj njegovog i zvršavanje. Neki od prekida ipak zahtevaju posebnu obradu. Recimo DOS ekstender obično reprogramira ju 8259 kontroler prekida tako da eksterne prekide 08H do 0Fh prebaci na neke druge radne brojeve kako bi mogao da ih razlikuje od prekida nastalih kao „General Protection Fault" „Stack Fault" i drugih. Najveći deo posla se ipak obavlje oko obrade prekida koji se i na će najviše koriste MS-DOS INT 21h, ROM BIOS INT 10h i sličnih.

Funkcije koje DOS ekstender mora obraditi možemo klasifikovati u četiri grupe:

- funkcije koje zahtevaju malo više od prebacivanje modova
- funkcije koje adresiraju bafera aplikacije ili podatka pa zahtevaju rad oko prebacivanja podataka i pre vođenje adresa
- funkcije koje u potpunosti moraju biti zamenjene da bi bile moguće u zaštićenom modu
- funkcije koje su jedinstvene za DOS ekstender. I prema tome omogućavaju izvršavanje zadataka koje MS-DOS ili ROM BIOS nisu u stanju da obave.

U prvu klasu funkcija spadaju pozivi gde su para metri smešteni u registrima, i ne obuhvaljaju rad se a drasama. Primer ovakvih poziva su DOS funkcije za u/izlaz/ulaz ne bezi karaktera (ekran serijski port, printer). Obrada ovih funkcija je jednostavna pošto prim zahtev se obradu DOS ekstender snima trenutno stanje zaštićenog moda prebacuje CPU u realni mod i ponovo ladaje zahtev za taj prekid fukciji za obradu relanog mode. Kada se obrada u realnom modu završi DOS ekstender vraća CPU u zaštićeni mod vraća sistem u prethodno stanje i vraća kontrolu e plikacija.

Druga klasa je nešto komplikovanija, s obzirom je od funkcije čiji parametri sadrže adrese nekih struk tura podataka. U originalnoj formi zaštićenog moda, ove adrese se DOS nemaju nikakvog smisla, i to iz dva razloga: adresa je u obliku selektor:ofset (dakle, selektori odgovara segmentu na neki način) a rezultujuća adresa je obično preko prvog megabajta. Ovaj problem DOS ekstender rešava korišćenjem internih bafera u konvencionalnoj memoriji kao privremenog skladišnog prostora. Na primer ukoliko se želi zapa nekog stringa na ekran DOS ekstender prasreće zahtev kopira string iz Data oblasti aplikacije (što je obič no u proširenoj memoriji) u bafer u konvencionalnoj memoriji formira novu adresu kojom zamenjuje origi nalnu adresu stringa prebacuje CPU u realni mod i šalje zahtev se lepa.

Treća grupa funkcija gde DOS ekstender zamenjuje funkcije potpuno novima se obično odnosi na rad se memorijom. Recimo funkcije 48h DOS preklda 21h koja alocira blok konvencionalne memorije se zamenjuje novom funkcijom kojom se alocira proširena memorija. Dok DOS i radi sa segmentima i paragrafima DOS ekstender operiše se selektorima tako da prosto preračunavanje nije moguće. Naravno za svaku nor malnu aplikaciju ovakva zamena je potpuno transparentna osim što odgovarajuće funkcije odjednom javljaju mnogo veće količine dostupne memorije nego u realnom modu.

Poslednje grupa funkcija su servisi koji su jedinstveni za DOS ekstender i obuhvataju preračunavanje realnih adresa u one za zaštićeni režim i obratno rad se deskriptorima i tabelama prekida, direktan pozivi prekidima u realnom modu i tako dalje. Većina ovih funkcije je namenjena internim potrebama i veoma specijalizovanim poslovima, i prosečna aplikacija i ne zna da oni postoje.

Očigledano je da su pisci prvih DOS ekstendera morali biti pravi virtuozi u programiranju ne samo zbog komplikovanosti posla nego i zbog potrebe da se bez prekorno pomaje način rada DOS-a, BIOS-a raznovr snog hardvera svih mogućih procesora na scen i tako dalje. Ne svu sreću, sa pojavom DPMI protokola posao

Podskup DPMI funkcija koje podržava *Windows* u standardnom modu

Prekidi i funkcije koje *Windows* DPMI ne podržava

Tabela 1

Broj	Naziv funkcije
0200h	Get Real Mode Interrupt Number
0201h	Set Real Mode Interrupt Number
0300h	Simulate Real Mode Interrupt
0302h	Call Real Mode Procedure with Far Return Frame
0303h	Call Real Mode Procedure with IRET Frame
0304h	Allocate Real Mode Call-Back Address
0305h	Free Real Mode Call-Back Address

Tabela 2

Prekid	Naziv prekida ili podfunkcije
Int 20h	Get Real Mode Interrupt Vector
Int 21h	00h Terminate process
	0Fh Open File with FCB
	10h Close File with FCB
	14h Sequential Read with FCB
	15h Sequential Write with FCB
	16h Create File with FCB
	21h Random Read with FCB
	22h Random Write with FCB
	23h Get File Size with FCB
	24h Set Realtive FCB Record
	27h Block Read with FCB
	28h Block Write with FCB
Int 25h	Absolute Disk Read
Int 26h	Absolute Disk Write

...sećamo DOS ekstendera je daleko olakšan posto je on taj preuzeo na sebe obradu većine hardverski zavisnih poslova, kao što je obrada prekida, prebacivanje modova i rad sa tabelama deskriptora.

WINDOWS – DOS EXTENDER I DPMI

Pojava *Windows 3.0* sistema (pažljivi čitalac je već primetio da insistiram na pojmu „sistem", a ne „program", ili što bi neki protivnici ovoga reći „zakrpa") je dovela do velikog proboja u oblasti rada sa zaštićenim modom ali i donela velike glavobolje proizvođačima DOS ekstendera. U *Windows*-ima DOS ekstender (čiji se naziv ovde nepravdano koristi) više nije potreban za svaku aplikaciju koja želi da radi u zaštićenom režimu već se nalazi u njima (njemu?) kao integralni deo.

DOS ekstender ugrađen u *Windows*-e se često bri ka se DPMI serverom. Za ovu zbrku ima i logično rešenje koje leži u onoj originalnoj specifikaciji koju je Ray Duncan čiso prvobitno se DPMI sastojao iz dva nivoa. Prvi nivo su činile funkcije niskog nivoa, na primer za rad sa prekidima prebacivanje modova i rad sa proširenom memorijom dok je drugi viši nivo obezbeđivao postojanje važnijih DOS ROM BIOS i miš funkcija u zaštićenom režimu. Tako definisan DPMI je zapravo verzija DOS-a u zaštićenom režimu, što je u vreme objavljivanja bilo svetogrđe. U već pomenutom „van pamćenom" dogovoru Microsoft je odustao od celog drugog nivoa funkcija prvobitnog DPMI protokola, ali ih kršom zadržao kao nedokumentovane. Sa druge strane, to je omogućilo DOS ekstenderima da nastave da žive a DOS sistemu produžilo život. Prema tome, DPMI kakav mi znamo danas, je pojačana verzija VCPI protokola, i njegove funkcije nisu zamišljene kao direktna podrška aplikaciji već DOS ekstenderu.

DOS ekstender ugrađen u *Windows* je kao programiranje rezidentnih programa pod DOS-om nikonezna tačno, ali mnogi znaju kako ipak nekoliko stvari je sasvim poznato. Pre svega, *Windows* u „enhanced" modu podržava kompletan skup funkcija DPMI 0.9 (koji je danas najzastupljeniji) dok u standardnom modu podržavaju samo mali podskup funkcija namenjen komunikaciji sa rezidentnim programima i device driverima (slika jedan). Dalje *Windows* podržava i skoro ceo skup prekida u funkcija MS-DOS API-ja osim onih na slici dva. Neke funkcije prekida 21h su delimično zastupljene a neke ni preterano bitne:

- INT 21h, funkcija 25h i 35h (uzmi i postavi vektor prekida)
- INT 21h, funkcija 38h i 65h („Get Country Info")
- INT 21h funkcija 44h podfunkcije 2,3,4 i 5 (IOCTL)

A ROM BIOS? Prvobitna DPMI specifikacija je eksplicitno nabrajala sve ROM BIOS pozive, ali ih nije sve podržavala, dok danas još uvek nije se sigurnošću poznato kako stoje stvari (u jednom dokumentu stoji: „Ukoliko aplikacija prenosi parametre registrima na korišćenje adrese, segmentne registre, pokazivače a stek parametre ta aplikacija ta morala da radi u zaštićenom modu *Windows-a*" - na sreću, miš, video, tastatura, printer i komunikacije spadaju baš pod one funkcije koje rade sa registrima).

U sledećem nastavku ćemo konačno početi se konkretnijim stvarima kako za DOS zaštićeni mod, tako i za *Windows* DPMI server.

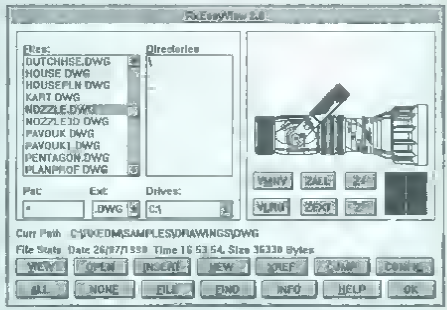
NA PRVI POGLED

Softverske slatke

Laki pogled iz Rasterax-a

Korisnici *AutoCAD*-a su oduvek poseza li za raznim DOS aplikacijama da bi brzo, bez njihovog mukotrpnog otvaranja, razgledali crteže na disku. *Rasterax* im ovo omogućuje bez napuštanja osnovnog programa.

Recimo da radite se nekim crtežom i želite da ubacite neki postojeći blok se diska. Pre verzije 12 *AutoCAD*-a, bili ste prinuđeni na mukotrpno kucanja dugačkih staza do crteža, nadajući se, pri tom, da nećete napraviti grešku u kucanju. Novi *AutoCAD* i upravljanje datotekama preko *dialog-box*-a olakšali su pretraživanja po disku. Međutim, problem i dalja ostaja ako korisnik ne može da se priseti ni približnog naziva datoteke.



Neka aplikacije, kao *AutoManager*, omogućavale su brz pregled crteža, bez čekanja karakterističnih za svako otvaranje – incijalizacija menija i slično). Problem se takvim programima je bio što se moralo izlaziti u DOS i potom vraćati u *AutoCAD*. Očigledno, *AutoCAD*-u je nedostajao sopstveni *preview*.

Norveška firma Rasterex specijalizovala se za izradu akceleratorskih kartica za CAD programe i aplikacionog softvera za njihovu podršku. Većina programa ima svoga pamjaka za VGA platforme, što je slučaj i se programom *EasyView*. Razvijen je kao ADS aplikacije za *Proteus* interfejs. Neuobičajeno za tako male aplikacije (oko 300 K), program radi isključivo uz hardversku zaštitu (*hardlock*).

Pored pomenute zaštite, u paketu dobijate uputstvo i jednu disketu. Posle izvršene instalacije, sve što je potrebno je da u ACAD.ADS ubacite RVIEW.EXP kao *autoload* aplikaciju, i *EasyView* je spreman za rad.

Kucanjem RXEV iz komandna linije, dobijate radni ekran, koji je podeljen na ekran za pregled, zonu datoteka i dugmad za menije. Sa leve strana nalazi se spisak datoteka u tekućem direktorijumu, se ekstenzijom DWG, SLD ili BAK („backup" crteža). Prosti m „klikom" na naziv datotake, ona biva obeležena i tada možete pozvati neku od operacije ispisanih na dugmadima:

- **View** izvršava prosto prikazivanje *preview* crteža u okviru zone za pregled. Crtež se pojavljuje veoma brzo, čak i ako ste konsnik VGA verzije, omogućujući vam dalje opcije. Preko dugmadi koja se nalaze direktno ispod ekrana, možete slobodno zumirati crtež i kretati se po njemu u svim pravcima. Pozivanjem opcije **vlrg** (*View Large*), crtež zauzima celu površinu *dialog-box*-a, čime dobijate mogućnost preskoka iz *Paper Space* u *Model Space* i obrnuto, kao i dodatne opcije zumiranja i panovanje crteža.
- **Open** inicira istoimenu *AutoCAD*-ovu komandu za otvaranje crteža čije ste ima izabrali, koristeći pri tom *Lisp* aplikaciju i *script* proceduru iz *EasyView* paketa.
- **Insert** ubacuje izabranu datoteku (ali samo crtež – slajdovi i bekapi nisu dozvoljeni) kao blok u postojeći crtež iz kojeg smo pozvali *EasyView*. Aplikacija prosleduja *AutoCAD*-u samo ime bloka i prepušta ubacivanje standardnoj proceduri komanda koja se može koristi za ovu svrhu.
- **Xref** izvršava sličnu operaciju, samo uz pozivanje **xref attach** komande, i datoteka se shodno tome tretira kao *external reference*.
- **Camp** od vas traži da odaberete datoteku koju želite da uporedite se onom koju ste već izabrali. Vizuelno poređenje vrši se tako što se pokreće **Large View** i oba crteža postavljaju jeden pored drugog.

U slučaju da ste odabrali više datoteka odjednom otvara vam se još nekoliko mogućnosti. Možete ih pregledati jednu po jednu uzastopnim pritiskom na **view** i biranjem **next** iz **Large View** moda, ili opcijom VMNY (**View Menu**), kada na ekranu dobijate brzi prikaz do 16 crteža koje ste izabrali.

U verziji 2.0 dodate su dve nove funkcije. Prva je poznati **File Find**, za lako pronalaženja datoteka po sistemu, a druga **Files** za upravljanja datotekama (brisanje, kopiranje ...).

U samom programu moguće je izvršiti i finu konfiguraciju parametara, kao što je korak pomeranja po crtežu, količina memorije koju dodeljujemo programu, rezolucija, inicijalni direktorijumi, pa čak i mogućnost de poliinije u crtežu vidimo u njihovoj originalnoj debljini (što će, naravno, uticati na brzinu prikazivanja).

Program *EasyView*, se korisničke strane, zadovoljava sve potrebe rada se datotekama, ubrzava i olakšava pronalaženja crteža. Možda će cena od 430 DEM izgledati preterana konsnicima naviknutim na *Automanager*, ali je ona, s obzirom na ozbiljnost pristupa (*hardlock* i detaljno uputstvo), ipak opravdana.

Vladan Aleksić

ČEKAJUĆI NA VEZU

Upravijanja modamima gledano iz ugla programara koji piša komunikacini program za automatsko odgovaranja i obradu poziva korisnika daleko je od jednostavnosti na koju ukazuju uputstva za njihovu primenu.

Zoran Životić

Kao što sam pokuzao u pretpročlom broju „Računara", problem nije toliko u procedun koliko u različitom ponašanju različitih modela modema u istim situacijema. Poseban problem su i sitne nedoslednosti u odnosu na deklarisano ponašanje modema, koje neiskusan programar u ovoj oblasti može da plati satima izgubljenog vremena.

U prethodnom tekstu bilo je reči o načinu inicijalizacije modema i odgovaranju na poziv. Pitanje prepoznavanje trenutka uspostavljanja veze i brzine komunikacije takođe namaće probleme slične onime o kojima je bilo reč

GDE SE DENUG KERIJER

Prvo treba da ustanovimo generalni metod kojim ćemo moći da utvrdimo da li veza sa drugim modemom postoji ili ne. Ovaj zadatak je trivijalan, jer modem ka računaru postavije odgovarajući status bit, koji označava postojanje *carrier* signala – signala koji postoji samo u toku modemske veze. Dakle, jednostavnim ispitivanjem ovog bita možemo tvrditi da li veza postoji ili ne:

```
int on_line(void)
{
    return com_status() & CARRIER;
}
```

Kada program prepozne da „telefon zvoni" i kada pošalje modemu naredbu za podizanje slušalice („ATA"), ostaje mu da uđe u petlju u kojoj će određeno vreme čekati pojavu *carrier* statusa i, ako se to desi, smatrati da je veza uspostavljena i da može konačno da počne da obavlja svoj pravi posao komunikacije sa korisnikom

```
// prethodna petlja za čekanje na poziv
// i podizanje slušalice

andat = time( NULL ) + WAIT_FOR_CONNEC
TION

for(;;)
{
    if ( on_line() )
        return CONNECTED;
    if ( time( NULL ) > andat )
        return FAILED;
}
```

Šta se dešava za vreme WAIT_FOR_CONNECTION i koliko ono treba da bude? Za to vreme se dešavaju razne stvari u zavisnosti od tipa modema – od prostog pregovaranja o brzini komunikacije do pregovora o protokolu komunikacije (MNP, V42 itd.), tako da ono može biti od svega nekoliko sekundi do 10, pa čak i 15 sekundi kod modema koji koriste protokole za korekciju grešaka. Iako bi komunikacioni program mogao da proće i sa ovako trivijalnom petljom, ona je uvek nešto složenija iz nekoliko razloga.

Prvo, pitanje je šta kada istekne period WAIT_FOR_CONECTION – šta uraditi sa modemom u tom trenutku? Modem može biti u jednom od dva stanja – ili još uvek pokušava da uspostavi vezu, ili je već odustao i poslao računaru poruku „NO CARRIER" Vreme koje modem pokušava da uspostavi vezu se programira registrom S7 modema (ATS7=x naterža modemu da o uspostavljanju veze pregovara x sekundi i da potom odustane) pa konstantu treba birati u skladu sa ovom vrednošću. Najjednostavnija varijanta je podesiti S7 na period veći od intervala koji je izabran u programu, tako da modem u trenutku kada program želi da odustane još uvek drži „podignutu slušalicu". Tada prosto izvodimo funkciju za prekid veze koju ćemo inače koristiti za tu namenu, i vraćamo se na početak – inicijalizaciju modema, čekanje na novi poziv itd.

Drugo, pitanje brzine komunikacije o kojoj su se modemi dogovorik može biti od značaja za program iako ne u toj meri u kojoj se to čini na prvi pogled. Naime, svi savremeni modemi koji imaju ugrađen neki od protokola za korekciju grešaka u stanju su da vrše takozvani *speed buffering* – da brzinu komunikacije sa računarom obsvijaju pri brzini koja je konstantna, veća, i nezavisne od stvarne brzine komunikacije na liniji. Konkretno, kod 2400 MNP modema brzine komunikacije računar-modem se uvek postavija na 4800 ili 9600, a pri tome sama veza može biti 2400 bps ea ili bez MNP-a ili bilo koja od drugih manjih brzina koju modem podržava. Konkretna posledica je da se brzina komunikacionog porta može isprogramirati na konstantnu vrednost i potom na nju potpuno zaboraviti – modem će o svemu sam voditi računa.

Na starijim modemima bez protokola za korekciju grešaka (i noviji mogu da rade na isti način, ali je potrebno izabrati poseban režim) bilo je potrebno brzinu komunikacije računar-modem podesiti prema stvarnoj brzini komunikacije na liniji nakon što se veza uspostavi. S obzirom da su takvi modemi već ugiavnom prošlost, ovom pitanjem se na bemo morali baviti da nije drugog razloga – poznavanje stvarne brzine veze može biti značajno za sam program pri proceni vremena potrebnog da se neka operacija završi. Tipičan primer je prenos datoteka – program može da odbije prenos ukoliko ustanovi da će trajati duže od vremena koje korsnik ima na raspolaganju za jednu sesnsu.

Nakon što postavi CARRIER status na komunikacionom portu, modem šalje računaru poruku iz koje se može saznati brzina komunikacije. Ova poruka kod svih modema sadrži reč CONNECT, dok je sam format poruka različit. Najčešće je to samo CONNECT xxxx gde je xxxx brzina komunikacije ali je poruka kod modema se ugrađenim protokolima često znatno duža pa i sastavljena iz više redova. Način na koji ćemo ovu poruku primiti je identičan onome što smo već imali kod inicijalizacije – jednostavno u petlji primamo znak po znak i sastavljamo poruku do prijema znaka CR, tako da ovde taj fragment koda neću ponavljati

Pri konfigurisanju modema i analiziranju CONNECT poruke treba biti posebno pažljiv. Kod savremenih modema brzina koju će prijaviti može se konfigurisati tako da buda ili stvarna brzina komunikacije na liniji ili brzine komunikacije računar-modem. Kod nekih modema se CONNECT poruka uvek odnosi na brzinu računar-modem, a posebne CARRIER xxx poruke prijavljuje stvarnu brzinu veze. Zbog svega toga važno je da program omogućava konfigurabilnost u prepoznavanju poruke. One se može izvesti na više načina, recimo upisivanjem svih mogućih poruka koje modem u trenutku uspostavljanja veze može da vrati, i upanvanjem ovih poruka sa odgovarajućim brzinema. Druga varijanta je neka vrsta analizatora poruke, gde dajete format poruke i mesto na kome treba potražiti broj koji se odnosi na brzinu komunikacije

Od trenutka postavljanja CARRIER statusa na komunikacionom portu pa do prijema kompletne CONNECT poruke takođe protekne neko vreme koje sa neznatno razlikuje od modema do modema, najčešće oko 2-3 sekunde. I ovde sa može desiti da ono što primimo nije samo čista CONNECT poruka već se tu može umešati i u baferu zaostala RING poruka (ako se „podizanje slušalice" bazira na analizi RING linije a ne RING poruka), pa sve što je rečeno kod sastavljanje odgovora modema na inicijalizaciju važi i ovde

U svakom slučaju, čak i ako ne želimo da uopšte analiziramo CONNECT poruku, potrebno ju je na nekn način „pojesti". U suprotnom, program će krenuti da traži od korsnika njegova ima a primiće ovu poruku koje je zaostale u baferu. Zato treba obavezno omogućiti konfigunisanje vremena sastavljanja CONNECT poruke i nakon isteka tog panoda potpuno očistiti prijemni bafer – tek tada program može početi normalno da radi

VEZA MOŽE DA POČNE

Konačno smo stigli do trenutka kada program nema posebnih briga sa modemom – jedino što treba da čini je da povremano proverava da li veza i dalje postoji. Funkciju za

tu namenu (**on_line()**) smo već videli. Ipak, ako ova funkcija treba da poslužiti kao generalna funkcija koja proverava stanja veze u svakom trenutku i sa svim modemima, one mora biti za nijansu složenija. Naime, u toku rada zbog raznih smetnji na vezi dolazi do povremenog gubitka *carrier* signala i tada modemi konsultuju sadržaj registra S10 – ukoliko je trajanje gubitka signala duže od isprogramiranog perioda, modem definitivno prekida vezu (neki modemi kao period koriste različiti vrednosti registara S9 i S10, ali to nije preterano značajno za ovo razmatranje). Ono što se iz uputstva za modem ne može saznati je šta se tada dešava sa CARRIER statusom koji modem postavlja na računarskom komunikacionom portu u ovim situacijama. Kod najvećeg broja modema ovaj status ne prati stvarno stanje *carrier* signala niti njegove lokalne padove – modem drži status sve do trenutka dok na zaključi da je period gubitka suviše dug i tada prekida vezu. Kod nekih modema situacija je potpuno drugačija – modem postavlja status tačno onako kako vidi *carrier* signal, što znači da i računar vidi sve lokalne padove. Zato se može desiti da funkcija **on_line()** primeti i neki kraći gubitak *carrier*-a, iako taj gubitak neće stvarno nateratí modem da prekine vezu.

Rešenje je jednostavno – uzorak CARRIER statusa treba uzimati u više pokušaja, i tak ako svi daju negativan rezultat, programu vratiti status o konačnom gubitku veze

```
int on_line(void)

{
    time_t endar,
    endar = time( NULL ) + CARRIER_LOST_TIMEOUT;

    for(;;)
    {
        if ( com_status() & CARRIER )
            return 1;
        if ( time(NULL) > endar )
            return 0;
    }
}
```

Period CARRIER_LOST_TIMEOUT se može prilično slobodno izabrati kao period od jedne do dve sekunde jer kada *carrier* stvarno nestane, nema načina da se ponovo pojavi do novog poziva, što se može desiti tek nakon znatno dužeg perioda.

PREKID VEZE

Ukoliko vezu prekine modem sa druge strane onda će ovu situaciju uočiti funkcija **on_line()** i program tu nama posebnog posla – modem će sam preći u komandni režim i program može da se vrati na početak – inicijalizaciju i čekanja na novi poziv. Ako, pak, konsnik izvodi programom predviđenu proceduru odjavljivanja, onda je program taj koji vezu mora da prekine. Uputstvo za modem predviđa jednostavnu proceduru – treba napraviti kraću pauzu (u smislu da se za to vreme modemu ne pošalje ni jedan znak), zatim se modemu pošalju tri znaka + (plus) i nakon ponovne kraće pauze modem prelazi u komandni režim kada mu se može poslati naredba ATH0 kojom se veza prekida.

I zaista, ako iz komunikacionog programa u toku veze sa drugim modemom pokušate ručno da izvedete ovu proceduru, ustanovićete da ona savršeno funkcioniše – nakon kucanja tri uzastopna znake plus, modem javlja OK i možete otkucati ATH0 komandu. Zatim to isto uradite programski

```
void disconnect(void)

{
    pause( EGT );
    com_puts("+++");
    pause( EGT );
    com_puts("ATH0\r");
}
```

I...od prekida veze ništa. (konstanta EGT je takozvani **Escape Guard Time** i treba ja podesiti prema periodu koji je podešen na modemu registrom S12). Kod velikog broja modema ova procedura jednostavno ne radi. Ono što je interesantno je da najveći broj konsnika modema i u svojim komunikacionim programima ime konfigurisan string za prekid veze potput „+++ATH0„M", koji uopšte vezu na prekida. Prekid se dešave zbog jedne sasvim druge opcije, o kojoj će biti nešto kasnije reči.

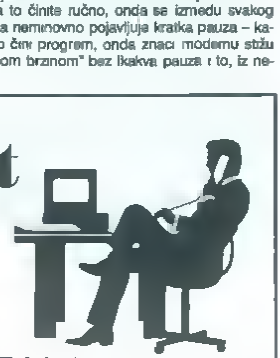
Problem je, kako ja praksa pokazala, u brzini kojom se ona tri znaka +++ šalju modemu. Kada to činite ručno, onda se između svakog znaka nemenovno pojavljuja kratka pauza – kada to čini program, onda znaci modemu stižu „punom brzinom" bez ikakva pauza i to, iz nekog nedokučivog razloga, tera veliki broj modema da ih na tretiraju kako specijalnu sekvencu bez obzira na dužinu pauze pre i posle. Zato je potrebno između slanja svakog od znakova + ubaciti kratku pauzu – dovoljna je i 1/10 sekunde – i veza će konačno moći da se prekine i na ovaj propisani način.

Konščenja ovog metoda nameće i jedan problem koji je više pitanja elegancija nego ozbiljan problem. Program koji opslužuje pozive korisnika mora, za razliku od običnog komunikacionog programa, da svaki primljeni znak vrati nazad, kako bi korsnik uopšte video šta radi. Ako korsnik otkuca sekvencu +++ po opisanoj proceduri, ona će biti prosleđena modemu kao da je lokalno otkucana – modem tada prelazi u komandni mod. Iz ove situacije će modem lako izaći ako korsnik sa druga strane otkuca sekvencu kojom želi ponovni povratak u režim prenosa podataka ili prekine vezu. Ipak, program nema načina da sazna šta ga je snašlo pa može nastaviti da šalje podatke korsniku koje će sada modem interpretrati kao komande. Sve to verovatno neće imati nikakve posledice po funkcionisanje sistema, ali bi jednostavno bilo elegantnije da se takva mogućnost potpuno spreči. To se može postići predefinisanjem znaka + u neki znak iz oblasti kontrolnih kodova koja program neće vratiti korsniku u toku normalna veza (ASCII kod se podešava registrom S2) i koji će na neki način ostati rezervisan samo za ovu namenu.

Postoji i alternativni način prekida veze koji se u praksi pokazao kao prilično pouzdan, iako ga *Hayes* kao tvorac standara komandnog seta modema ne preporučuje. Naime, kontrolna linija na komunikacionom portu sa oznakom DTR (*Data Terminal Ready*) ima veoma važan uticaj na rad modema. Ako DTR liniju oborite, modem postaja neaktivan za „apoljni svet", iako mu i dalja možete normalno sleti naredbe. Obaranje DTR linija u toku veze može imati čsbn dejstva, u zavisnoti od toga kako je modem podešen. Nakon naredbe AT&D0 modem potpuno ignoriše DTR liniju, nakon AT&D1 obaranja DTR signala prebacuje modem u komandni režim (u toku veze isti efekat kao i +++ sekvenca) i, konačno, nakon AT&D2 i AT&D3 modem pri obaranju DTR signala prekida vezu, uz dodatni interni reset, ako je zadata poslednja naredba. Prekid veza na ovaj način ja zato za program najjednostavnije rešenje.

Sve u svemu, na kraju bi se mogao izvući zaključak da kontrolisanje modema od strane programa nije nimalo težak zadatak – ono što ga čini donekle problematičnim ja različitost u ponašanju pojedinih modela. Posle više godina „druženja" se različitim modemima, moj je utisak da se razlike sve više smanjuju i da će verovatno i one najsitnije vremenom nestati. Do tada, ostaje vam samo da se držite pravila koga sam i ja počeo da se pridržavam čim sam svoj prvi modem zamenio novim – ne izvlačite nikakve zaključke o ponašanju modema na osnovu iskustva sa vašim modemom! Obavezno proverite bar sa još jednim modelom i za sve parametre koji se razlikuju odmah ostavljajte mogućnost da se konfigurišu. Samo tako će svaki novi model na koji naiđete donositi sve manja i manja iznenađenja.

ProSoft
34000 Kragujevac Ul. JNA 63
Phone/fax (034) 212 906
software
računarske mreže
projektavanje i oprema
SA NAMA I POSLE DVEHILJADITE



OBJEKTI, TABELE, INDEKSI

Ako ste, čitajući prethodne nastavke iz ove serije, formirali predstavu o načinu rada sa poljima i slogovima, onda vam nece biti teško da savladate poslednje prepreke do konačnog cilja: rad sa BLOB-ovima, tabelama i indeksiranje.

Miljan Jovanović i Vladimir Klisić

BLOB

U prethodnim nastavcima pokazali smo vrste polja i njihove tipove. Tom prilikom smo samo pomenuli posebnu vrstu polje predvidenu za smeštanje velikih podataka koji sa lnače ne bi mogli smestiti u nako polje „standardnog" tipa. To je slučaj sa slikama grafikonima ili dužim opisima (tekstovima) koji predstavljaju jednu celinu i koje treba lako čuvati. U drugim programskim paketima ste verovatno sreli posebnu vrstu polja koja izlazi na kraj sa takvim podacima – to su tzv. *memo* polja. U Borlandovoj varijanti ovakva polja nose naziv *BLOB*, što je skraćenica od *Binary Large OBject*. Uvodeći BLOB-ova Borland je želeo da ode korak dalje od memo-polja. Pogledajmo kake je cela koncepcija BLOB polja zamišljena.

Ne samo zbog veličina već i zbog činjenice da je la veličina potpuno proizvoljna za svaki pojedinačni slog jasno je da se smeštaj BLOB polja ne može čuvati unutar tabela gde su svi slogovi iste duzine iz istih razloga nija moguce n indeksiranje po ovoj vrsti polja. Kao posledica svega toga BLOB polje u slogu nije ništa drugo do pokazivač na podatak koji se čuva u posebnoj datoteci. Dakle n pitanju je vrlo jednostavan princip u tabeli sa na mastu BLOB polja nalaze osnovne informacije o lome gde je podatak smešlan koje je dužina i sl. Ti osnovni podaci zanzimaju 10 bajtova i njihova struk lnra je za nas nebitna ali na to moramo racunati. (Npr ako kre ramo polje nekog tipa *M40* ono ce zauz meti zapravo 50 bajtova n slogu.) Kad vec govorimo o datoteci sa BLOB podacima recimo i to da ona nosi isto ime kao i tabela nz koju je kreirana (clan familije) a ima ekstenziju *MB* (od *Master Blob*). Nazad bnduci da sam sadržaj BLOB polja uje n slogn za lakve polje nece biti od koristi većina PX lunkcija koje lnace rade sa poljima i slogovima, ali su umesto njih olezbedene posebne funkcije slične namene.

Zavirimo malo iza onoga što BLOB-ovi predstavlja ju Struktura MB datoteke se zasniva na nizu sektora iste dužine. Već na prvi pogled jasno je šta se na ovaj način postiže: brisanjem BLOB-a oslobađa se mesto novom. Kod drugih jezika se često baza mora „prepakovati" da bi se prostor fizički oslobodio (najčešce kopiranjem u pomoćnu datoteku samo on h delova koji nesu obrisani) jer se n lom slučaju ova datoteka formira prostim „lepljenjem" novih podataka na kraj. Brisanje se zapravo na postavljanju oznake da je data nformacija obrisana, ali taj prostor ostaja „zarobljen". U osnovi problema je nepoznata (promenljiva) dužina polja koja se upisuju – kako znati da necemo prebrisati neki validan podatak koji sled odmah iza? Kod BLOB ova lo n je slučaj. MB datoteka se sastoji od sektora. Svak sektor predstavlja jednu log čku celinu. Ona n osnovi ima sa mo dva stanja slobodne ja ili zanzata. Poznate je duzine i moguca je „planiranje" rasporeda podataka. No princip sektorisanje datoteke ne deje uvek povoljne rezultate: podatak kraći od sektora uzima ceo sektor pa ako je on duž ne 200 bajtova onda će ostalo (npr do 4k) bili prazno. U svakom slucaju na to se ovde ne treba obazirati – sve je potpuno transparentno i za programera i za konsnika. (Na postoji funkcija kojom se može direktno promeniti veličina sektora.)

Tipovi BLOB-ova

Rnkovodeni potrebama, tvorci *Paradox*-a su zamislili nekoliko podvrsta BLOB polja od kojih su n verziji *Paradox Engine 3.0* realizovana samo dva: polja n kojima se čuva neformatiran tekst (tip *M*) i polja u kojima se čuvaju neformatirani binarni podaci (tip *B*). Postoje i polja n kojima se čuvaju formatirani podaci a razlika izmedu jednih i drugih je u lome šlo kod formatiranih polja porad pokazivača na podatak postoji još zaglavlje koje se čuva *u slogu* i koje na neki način opisuje strukturu podatka na koji referše. Ako je reč o formatiranim tekst BLOB polima (tip *F*) onda zaglavlje može sadržati informacije n lome kako je sam tekst formatiran (margi ne broj redova po stranici) ako je (drugi primer) podatak n polju neki ikoni onda zaglavlje može sadržati informacije o njegovoj strukturi. Formatirani grafički BLOB (tip *G*) se od BLOB a tipa *B* razlikuje samo po lome sto ima zaglavlje. Ovaj BLOB tip je kako mu samo ime kaže predviden za čuvanje slika – u zaglavlju se čnve ojas formata n kojem je slika. U upotrebi je još i OLE BLOB koji služi kao podiška *Windows* OLE koncepta („Računari 87" dec 92). Na zalost formalirana polje *Paradox Engine* trenutno ne podržava već samo *Paradox* za *Windows*. To znači da *Paradox Engine* trenutno ne može da skoristi informacije sadržanm u zaglavlju bar ne preko postojecih lnnkcije. Naravno pul vam je otvoren da ovaj nedostatak popravite pisanjem sopstvenih

Iako smo rekli da neformatirani BLOB-ovi nemaju zaglavlje ipak je omoguceno da se prvih *nnn* (0-255) bajtova nekog BLOB a može čuvati i unutar sloga. Za lako nešto često ima potrebe. Vodeci deo je kopija prvih *nnn* bajtova u BLOB datoteci i može biti kratak ključni opis podatka koji predstavlja. Tako umesto da lnažite neki BLOB po MB datoteci vi samo pročitate njegov vodeci deo koji je u slogu i na osnovu toga zaključite da li sta ga pronašli ili ne. Dakle kada specificirate BLOB polje sa *Mnnn* ili *Bnnn* vi definišete neformatirano BLOB polje koje ima početnih *nnn* bajtova prekopiranih u slog. U situacijama kada je ukupna veličina (aventualnog) zaglavlja – tela samog BLOB-a dovoljno mala da stane cela u specificiranih *nnn* bajtova. *Paradox Engine* će čitav BLOB staviti u slog ne koristeći prostor n MB datoteci. Ovo se odvije potpuno neprimetno za konsnika i predstavlja neku vrstu interne optimizacije.

Čitanje sadržaja BLOBa

Pogledajmo sada kako se odvija tipičan rad sa BLOB polima. Zbog lanje pomenutih ograničenja prilkom čitanja nekog sloga na čita se BLOB polje već samo pokazivač na njega i njegov eventualni vodeci deo. Čitanje vodećeg dela obavlja lnnkcija **PXBlobQuickGet** (manjah isključ vo ovaj vodeci deo nije moguce). Medu lim da bi se pročitao neki podatak iz MB datoteke, potrebno je uraditi sledece

1 Pročitati slog (**PXRecGet**) kako bi se došlo do inlormacije o polima
2 Pozovati funkciju **PXBlobOpenRead** koja otvara MB datoteku za citanje i pozicionira se u njoj ne osnovn id bioja sloga i id broj BLOB polja. U program se vraca id broj pomoću kojeg se dalja dnik tno prislupa podatku (BLOBHANDLE)
3 Pozovati funkciju **PXBlobGet** koja učitava najvše do 65520 (FFF0h) bajtova n jednom prolazu. Dvo ograničenje je posledica DOS ove alntekluie. Da bi se pročitao podatak čija veličina prelazi pomenutih FFF0h potrebno je vise puta pozvati lnkciju **PXBlobGet**, svaki put sa novm olsetom od kojeg treba početi čitanje. Razume se pre bilo kakvog čitanja treba obezbediti prostor u memoriji u koj će se podatak smestiti

Mogla je sama koncepcija čitanja „od ofseta ne zgodna za rad i treba malo vremena da se programski prnlkne na nju. Zamerka je sledeća: ako vec čitaramo BLOB za čitanje pasnja na sličan način kao obicnu datoteku na disku zalto onda za prstup podatku treba dava i ofset? Zar nije logičnije da postoji neki (interni) pokazivac na tekucu pilaciju koji će bi ažuriran prilkom svakog čitanja ili pisanja? To je još jedna stvar koja se kosi sa tradicijom ali je na srecu bezazlena.

4 Pozvati funkciju **PXBlobClose** koja zatvara dati BLOB određen svojm ID brojem. Drugi parametar ove funkcije u ovom slucaju nije od značaja (ij bez obzira na vrednost smatra se da je **PXBLOBACCEPT**)

Upisivanje i promena sadržaja BLOB-a

Potnpak oko upisivanja sadržaja BLOB polja zaslu žuje malo veću pažnju.

1 Čita se slog (**PXRecGet**) kako bi se došlo do inlormacija o poljima
2 Otvara se BLOB za upisivanje. Ovde mogu nastati dva situacije
   A BLOB već postoji n MB datoteci i treba mn pristupiti da bi se izmenio
   B BLOB se prvi put popunjava (nema ga u MB datoteci)

U oba slučaja su isle pripramne radnje. Poz va se funkcija **PXBlobOpenWrite** koja odvaja prostor u koji ce se upisivati sadržaj. Taj prostor vidimo kao neku naš privatni BLOB koji dobija svoj sopstveni BLOBHANDLE i koji je potpnno nevidljiv za avantualna ostala konsnike. Za taj privatni BLOB se može alocirati od 0-256MB što se prosleduje kao argument prilkom poziva funkcije **PXBlobOpenWrite**. Eventualni BLOB u MB datoteci ostaje netaknut.

Odavde nastaju razlika. Ako zelite da sačuvate stan sadržaj BLOB a koji je n MB datoteci treba da funkcije **PXBlobOpenWrite** lo naznačite ako njen poslednji argument bude **PXBLOBCOPY**. n alocirani prostor ce se upisati postojeći sadržaj BLOB a. Ako je taj prostor kojim slučajem manj od veličine BLOB-a koji treba da se upiše prekopiraće se onoliko bajtova koliko može da stane. U slučajn da nas ne zanima postojeći sadržaj BLOB-a poslednji argument treba da bude **PXBLOBNEW**. I tada se u upravo kreiran prostor nece kopirati ništa

3 U privatni BLOB se upisuje novi sadržaj. I upis je ograničen na 65520 bajtova u jednom protazu tako da se za veće podatke upis vrši iz više prolaza po sličnom mehanizmu kao pod čitanja (svaki put se zadaje novi ofset od kojeg počinje pisanje)
4 Ključni trenntak odluka o tome sta ce se nradili sa BLOB-om. Poziva se funkcija **PXBlobClose** kod koje je ovaj pul argument koj se prosleduje uz BLOB HANDLE i te kako ima znacaje. U zav snost od nje gove vrednosti može se dogoditi sledece
   A. Argument ima vrednost **PXBLOBACCEPT** privatni BLOB zamenjuje onaj javni (*public*) iz MB datoteka na taj način što se ID broj javnog BLOB a poslavlja na vrednost ID broja privatnog BLOB-a. Sam sadržaj ce biti upisan u MB datoteku prilkom poziva bilo koje od lnnkcija **PXRecAppend** **PXRecUpdate** i **PXRecinsert**. Velic na BLOB a je sada ona koja ste zadali prilkom poziva lnnkci je **PXBlobOpenWrite**
   B. Argumenti ima vrednost **PXBLOBREJECT** privatni BLOB *nece* zamenili postojeći javni BLOB

Ruku ne arce sam mehan zam upisivanja BLOB-a n MB daloteku ja n stvarnosti malo drugačij mada je nevidljiv za konsnika. Privatni BLOB se odmah upisuje u MB daloteku (npr nedge na kra) ali još uvek nije označen kao validan. Ažuriranjem sloga se samo proglašava za važeći. Za nas kolisn ke je on negde izvan MB datoteke. Ovo sve ne mora da nas zanima. o pi vatnom BLOB-n možemo razmišljati kao o zasebnoj celini

Iz ove priče mogli ste zaključiti kakva je razlika izmedu privatnih javnih BLOB-ova. Javni BLOB-ov su on koji su u MB datoteci i koj ma mogu pristupati svi eventnalni konsnici pomoću funkcije **PXBlobOpenRead**. Privatni BLOB-ovi se pojavljnju samo pul kom poziva funkcije **PXBlobOpenWrite** i nj ma ne mogu pristupiti drugi konsnici os ni vas

Pomocne funkcije

Pomenimo još neke funkcije za rad sa BLOB-ovima. BLOB se može obrisati funkcijom **PXBlobDrop** kojom se eventualni privatni BLOB-ov odmah brišn a javn n BLOB se brise tek prilkom upisa sloga u tabelu za koju je vezan. Ekvivalentan afekat se dobija funkcijom **PXPutBlank** (primenjanom na BLOB-polje) lk funkcijom **PXRecBufEmpty** (pomenjenom na slog n kojem ima BLOB polja)

Listing 1

```
#include <stdio.h>
#include <pxengine.h>

int main(void) {
    PXCODE PXErr;

    char *Table= "table";
    FIELDHANDLE PrimaryIndex=0;
    FIELDHANDLE SekundarniIndeks[255];
    FIELDHANDLE IncSekundarniIndeks[255];

    // pokretanje sustava
    PXInit();

    // dodavanje primarnog indeksa
    PXErr= PXKeybyId(Table, 1, PrimaryIndex, PRIMARY);
    printf("AddKey [PRIMARY] %s\n", PXErrMsg(PXErr));

    // dodavanje sekundarnog indeksa bez održavanja
    PXErr= PXKeybyId(Table, 2, OnemogućeniIndeks, SECONDARY);
    printf("AddKey [SECONDARY] %s\n", PXErrMsg(PXErr));

    // dodavanje sekundarnog indeksa sa održavanjem
    PXErr= PXKeyAdd(Table, 3, IncSekundarniIndeks, NCSECONDARY);
    printf("AddKey [INCSECONDARY] %s\n", PXErrMsg(PXErr));

    // gašenje sustava
    PXExit();
    return(PXErr);
}
```

Primer kreiranja sekundarnog indeksa

Listing 2

```
#include <stdio.h>
#include "pxengine.h"

int main(void) {
    PXCODE PXErr;

    char *table= "table";

    FIELDHANDLE MapiranoPolje;
    FIELDHANDLE IdMap;

    // pokretanje sustava
    PXInit();

    // mapiranje novog polja
    PXErr= PXKeyMap(Table, 3, MapiranoPolje, Mapirano Polje, 1, &IdMap);
    printf("KeyMap %s\n", PXErrMsg(PXErr));

    // dodavanje sekundarnog indeksa na mapirano polje
    PXErr= PXKeybyId(Table, 1, &IdMap, INCSECONDARY);
    printf("AddKey [Map] %s\n", PXErrMsg(PXErr));

    // gašenje sustava
    PXExit();
    return(PXErr);
}
```

Primer mapiranja polja i kreiranja kompozitnog indeksa

Veličinu BLOB polja možete saznati pomoću funkcije **PXBlobGetSize** koja će vam vratiti stvarnu veličinu BLOB polja. Ako želite da sadržaj BLOB polja u nekom slogu bude isti kao i sadržaj BLOB polja u nekom drugom slogu, možete koristiti funkciju **PXBlobCopy**. Ako želite da sadržaj BLOB polja u nekom slogu bude isti kao i sadržaj BLOB polja u nekom drugom slogu, možete koristiti funkciju **PXBlobCopy**. Ako želite da sadržaj BLOB polja u nekom slogu bude isti kao i sadržaj BLOB polja u nekom drugom slogu, možete koristiti funkciju **PXBlobCopy**.

Funkcijom **PXBlobClone** možete kreirati svoje privatne kopije nekog javnog BLOB-a – za samo jedno BLOB polje ili za sva BLOB polja unutar sloga. Te kopije onda možete menjati i sa **PXBlobClose** ponovo ih javnim. Kada se pojavljuje potreba za ovom funkcijom? Recimo da ste učitali neki slog u slog bafer. Imate ID brojeve BLOB-ova. Dakle možete im pristupiti. Međutim, ako se u tabeli pomeri ne (recimo) sledeći slog, ili ID brojevi BLOB-ova nisu više isti kao sada upotrebljeni neki od **PXBlob** funkcija, vi decete BLOB-ove kao da su nulte dužine. Zato da biste sačuvali sadržaj BLOB polja pri ikom kretanja po tabeli, prvo ih iskopirajte na u svoje privatne ekvivalente sa **PXBlobClone**, a zatim se kretite po tabeli. Ako se nakon kopiranja ovih BLOB-ova pomen pokazivač sloga u tabeli njihov sadržaj neće biti promenjen. Ove kopije možete menjati i lako izmenjene pomoću **PXBlobClose** postaju u BLOB datoteku. Od log trenutka ti BLOB ovi postaju javni.

U visokom sistemu okruženju je situacija obično pažljiva sa BLOB polja malo složenija. Kao takva biće detaljno obrađena u okviru posebnog naslavka koji će se baviti konkurentnim *Engine*-a pod mrežom.

TABELE

U prvom članku iz ove serije smo tabele samo uzgred pomenuli a sada ćemo im posvetiti malo više pažnje. Rad sa tabelom podrazumeva ne samo pristup i ažuriranje datoteke sa ekstenzijom DB već i otvaranje različitih procesa koji se ne tiču oslabih datoteka u familiji (ažuriranje indeksa na primer).

Za rad sa tabelama su obezbeđene sve potrebne funkcije počev od funkcija za kreiranje DB datoteke preko funkcija za kreiranje indeksa pa sve do funkcija za kopiranje i brisanje čitavih tabela.

Kreiranje tabele

Pre samog kreiranja tabele treba obaviti neke pripremne radnje koje će specifi eventualne kasnije greške. Prva stvar koju treba da proverite je da li tabela već postoji na disku. Kreiranje tabele sa istim imenom bi nepovratno obrisalo postojeću tabelu a malo je verovatno da je to ono što želite. Da biste saznali da li tabela pod imenom „ABC0" već postoji na disku, uradite sledeće: pozovite funkciju **PXTblExist** i kao ime navedite „ABC0". Ako postoji adresa *int* promenljive koja je po povratku biti popunjena na nula vrednošću, ako ne postoji.

Pre kreiranja možete izabrati format u kojem će tabela biti sta i novi. Stari format je onaj koji je podržavan u verziji 3.5 *Paradox*-a (odnosno verziji *Paradox Engine 2.0*) a novi je najnoviji onaj koji kreiraju *Paradox 4.0* i *Paradox Engine 3.0*. Malo je zbunjujuće što ova dva pekata ne nose isti broj verzije ali to sad nije naš problem. Treba zapamtiti da oba postoje dve verzije tabela *3.5* i *4.0* tako se podrazumeva da se tabele kreiraju u formatu 4.0 ostavljena vam je mogućnost da ovo stanje po želji menjate u toku programa. Potrebno je samo da pre kreiranja konkretne tabele funkcijom **PXTblCreateMode** definišete podrazumevani format tabela (3.5 ili 4.0). Biraju verziju 3.5 formata tabela, gubite mnoge lepe stvari koje odlikuju PE 3.0 sekundarne indekse koji zahvaljuju više polja BLOB polja. U opštem slučaju nema posebnih razloga da ne koristite novi format tabela – osim ako je u pitanju kompatibilnost sa starim programima. Mada, i u slučaju da imate posla sa tabelom kreiranom u formatu 3.5 poslovi funkcija **PXTblUpgrade** koja služi upravo za konverziju starog formata u novi. Ova funkcija se može ugraditi u zaseban program koji će lokom instalacija vaše aplikacije možete izvršiti. Upgrade na novu verziju dalje nesmetano radi. Očigledno je da i ova funkcija ide u paketu sa **PXTblCreateMode** što treba da obezbedi kompatibilnost sa prethodnom verzijom.

Rešili smo golovo sve probleme koji prethode samom kreiranju tabele. Ostao je samo još jedan. Podešavanje njene maksimalne veličine. Ovaj postupak ima vazan praktičan značaj, svaka *Paradox* tabela se po smaliana kao sekvencijalna datoteka sastoji z maksimalno 65536 blokova. Blokovi su interna stvar PE a. Prosto potreban za smeštanje sloga se odvaja u blokovima a ne u bajtovima a razlog tome je brzina pristupa podacima. Zato vam je ostavljene mogućnosti da sami izaberete veličinu tabele a time posredno i veličinu blokova. Možete izabrati jednu od ove tri mogućnosti (odgovarajuća veličina bloka je navedena u zagradama): 64MB (1K) 128MB (2K) ili 256MB (4K). Ovaj posao obavlja funkcija **PXTblMaxSize**. Podrazumevana vrednost je 128MB.

Došli smo najzad i do kreiranja tabele. Kreiranje se vrši funkcijom **PXTblCreate**. I tom prilikom se zadaju njeno ime, podaci naophodni za opis njena strukture (ime je obično ime DOS datoteke (ekstenzija DB se podrazumeva) a zatim njegova polje broj polja koje će tabela imati kao i niza stringova (i dan sa naz vima polja u tabeli a drugi sa njihovim tipovima). Te osnovne podatke PE će sačuvati u zaglavlju same datoteke tako da kasnije možemo lako saznati njihov broj i nazive.

Otvaranje i zatvaranje tabele

Otvaranje tabela prethodi bilo kakvom pristupu njenim slogovima. Funkcija koja se poziva da obavi ovu operaciju je **PXTblOpen**. Ona očekuje sledeće parametre:

- ime tabele je ASCII string (dozvoljeno je navođenje čitave DOS putanje) podrazumeva se ekstenzija DB a ako je druga biće ignorisana.
- adresa promenljive u koju će **PXTblOpen** upisati ID broj koji dodeli tabeli kako je već rečeno taj ID broj će za sva dalja obraćanja tabeli.
- redni broj koji odredite kakav indeks treba otvoriti uz tabelu. Za primarne indekse i tabelu koja nije indeksirana treba navesti 0. Da bi se otvorili tabelu po nekom sekundarnom indeksu navedite ovde broj polja (1-255) po kojem treba otvoriti indeks. Najzad, ako otvarate tabelu po kompozitnom indeksu navedite ID broj koji ste dobili pomoću funkcije **PXKeyMap** i **PXKeyQuery** (taj broj ne bi trebalo da bude veći od 255). Po BLOB polja se ne može vršiti indeksiranje pa o tome vodite računa.

Vrednost koja određuje da li će se promene nad slogovima odmah zapisivati na disk (ne nula) ili će se naj pra smeštali u interni bafer (nula).

Dozvoljeno je otvaranje jedne iste tabele više puta. Tada ćete za svako otvaranje dobiti novi ID broj. Ova situacija je česta kada želite da pristupate tabeli po više različitih indeksa. Kako je ovo više vezano za tehnika indeksiranja, to ćemo ovaj postupak detaljno objasniti kasnije.

Još samo mala napomena: podsetite se funkcije **PXSetDefaults** (opisane u prvom članku iz ove serije) kako biste podesili parametre neophodne za otvaranje više tabela.

Zatvaranje tabele se vrši funkcijom **PXTblClose** koja ima samo jedan parametar ID broj tabele. Možete zatvoriti samo tabelu otvorenu sa **PXTblOpen**. Ako ste uz tabelu otvorili sloga baleta, onda će se automatski oslobo-diti (zatvoriti). Sv eventualni zaključani slogovi biće otključan (ukoliko ih nije još neko zaključao). Ako ste jednu tabelu otvorili po više ndeksa onda morate pozvati **PXTblClose** onoliko puta koliko je pozivana funkcija **PXTblOpen** svaki put sa odgovarajućim ID brojem.

Čitanje strukture tabele

Nisu ralke situacije kada treba da pristupite tabeli čiju strukturu unapred ne poznajete. Pomenuli smo da se te informacije čuvaju u samoj DB datoteci pa je sasvim razumljivo da su obezbeđene funkcije za njihovo čitanje. Cilj je da se iz svake tabele preuzmu kompletne informacije o njenoj strukturi – mena tipovi polja, broj slogova, broj polja koja su zahvaćena primarnim indeksom i na kraju njihova imena. Tabela mora biti prethodno otvorena. Na osnovu razulata ovih funkcija može se stvoriti kompletan uvid u neku tabelu. Iako da možete napisavih program za pregled tabela koje nisu usko vezane za vaš projekat. Nema boljeg primera za ovakav program od domaćeg PAX a čiji je autor Zoran M. Todorović iz Novog Sada.

U prethodnom članku smo naveli nekoliko funkcija koje su predviđene za ovakva ispitivanja to su **PXRecNFlds** **PXFldName** i **PXFldType**. Njima treba pridodati i funkcije **PXTblNRecs** – elo vam osnove za eksperimentisanje. Kako bi u pojednostavljenoj varijanti išao lok isp itivanja? Kao prvo. Iraba videti od koliko polja se sastoji neka tabela (za to se koristi **PXRecNFlds**) da bi se odvojila memorija potrebna za smeštanje podataka (maksimalno je potrebno 26+5 bajtova po polju – za ime i tip). Zatim se odrediju mena i tipov polja dale tabela funkcijama **PXFldName** i **PXFldType**. Koliko je polja pod primarnim indeksom može se saznati pomoću funkcije **PXKeyNFlds**. Broj slogova u tabeli se može saznati preko **PXTblNRecs**. Radi se samo o slogovima koji nisu obrisani – da ne bude zabune. Ijen tabela može biti dugačka i nekoliko megabajta a da sadrži samo jedan slog. Za kompletne

nje sliče o nekoj tabeli postoji i funkcija **PXTblName** ona vraća ime tabele za dati ID broj (potreba za njom nam nije baš najjasnije ali „neka se nađe")

Brisanje, pražnjenje, preimenovanje, kopiranje, dodavanje tabela

Zahvati koji se obavljaju nad celom tabelom su uobičajena stvar za situaciju kakva je (recimo) pakovanje baze *jer Paradox Engine* nema funkciju koje bi bile ekvivalenti *Clipper*-ovoj komendi PACK. Da bi se tako nešto napravilo potrebno je da se upotrebe funkcije koje rade se tabelom u celini

Upotrebite

PXTblEmpty – za pražnjenje tabele. Svi slogovi će biti obrisani i tabela će imati inicijalnu dužinu baš kao da smo je tek kreirali. Svi javni BLOB-ovi će biti obrisani

PXTblDelete – za brisanje *Paradox* familije. Ovo je destruktivna komanda koje ce bez upozorenja obrisati *sve* članove jedne familije (kao da ste kucali DEL <TblName> *)

PXTblRename – za promenu imena familije. Slično brisanju sve datoteke koje nose isto ime kao i data familija biće preimenovane (i ovde se može napraviti paralela RENAME <OldName> * <NewName> *)

PXTblCopy – za kopiranje cele familije. Ovom komandom se na disku formira kopija nake familije pod drugim imenom (COPY <TblName> * <NewName> *)

PXTblAdd – za dodavanje slogova jedne familije u drugu. Ova funkcija se koristi za „pakovanje" baze. Postoje i drugi načini da se to učini ali ako na disku ima v se slobodnog mesta nego što jedna familija zauzima onda se može primeniti sledeci algoritam: prvo kreiramo pomoćnu tabelu iste strukture sloge (kao tabela koju želimo da prepakujemo); zatim joj dodamo sve slogove iz originalne tabele. Na ovaj način dobili smo čistu tabelu koja više ne sadrži obrisane slogova, ostaje nam još da ot ginalnu tabelu obrišemo i da pomoćnoj damo odgovarajuće ime

Potrebe za pakovanjem neka tabele se javlja kad slogove često brišemo i dodajemo nove. Iad se slogovi srodni po indeksu nalaze razbacan u tabeli – u ovom slučaju se pakovanjem može ubrzali rad sa dužim tabelama. Drug slučaj je kad posle ob mn jeg brisanja trebo smanjit dužinu tabele – ovo i nije neophodno ako se kasnije očekuje upisivanje novih slogova.

Traba de male u vidu da *Paradox Engine* prilikom poziva ovih „globalnih" funkcija obavlje mnoštvo za vas transparantnih operacija. To se posebno odnosi na rad u višekorisničkom okruženju kada se ovakve operacije nad celom tabelom ne mogu izvesti dok god im drugi korsnici pristupaju. Koje su to „nevidljive" operacije i šta se sve dogada u mrežnom okruženju otkacete u sledecem nastavku

INDEKSIRANJE

Na stranicama ovog lista više puta je pisano o tome šta su indeksi i čemu služe pa necemo l m problemom opterecivati i ovaj tekst. Umesto toga videcemo kako *Paradox Engine* kreira i upotrebljava indekse kao i kakva ograničenja postoje pri radu sa nj ma. Ako se seća te uopštene priče oko indeksiranja iz prvog nastavka onda znate da su u varziji PE 2.0 postojali samo primarni i sekundarni indeksi ta de im se u poslednjoj verziji pridruz o i kompozitni (mapirani) indeks koji je u suštini verijanta sekundarnog indeksa. Sve in vrste indeksa či ne mocno ali na želosť ne i kompletno rešenja za problema pratlaž vanja i sortiranja podataka. Kako cemo videti cela koncepcije je dobro zamišljena i polovično realizovana. Delovi koji su urađeni urađeni su dobro ali su ostali neki neresenti problemi posebno u vezi se pretraživanjem. Dok očekujemo novu varzju pogledaj mo šte nem pruže postojeća. Najpre reč-dve o tipovima indekse

Primarni indeks se formira na osnovu ključeva – polja koje *jednoznačno* odreduju svaki slog u tabeli. Otud se ne mogu duplirati slogovi koji imaju iste ključeve. Koliko ce polja činiti ključ (a ime l primarni indeks) stvar je projektanta baze. Za PE je b tino samo da la polja budu na početku sloga jedno uz drugo. Tako se primarni indeks moze kreirati na osnovu samo prvog polja. Ili prvog i drugog zajedno. Ill na osnovu prva ih polja ili ne može bil kre ian na primer od prvog drugog i četvrtog polja. Tabela se smatra indeksiranom samo ako postoji primarni indeks (sve poruke tipa „Table is not indexed" se odnose na njega!). Nakon kreiranja primernog indekse na disku ce se pojav ti datoteka istog naziva kao i tabela a sa ekstenzijom PX. Mala veličina ove datoteke je ono što posebno impresionira

Sekundarni indeks se kreira na osnovu samo jednog polja u tabeli pn čemu je položaj tog polja proizvoljen (moze biti čak i neko polje koje je deo primarnog indeksa). Ovo ne znači da može postojati samo jedan sekundarni indeks već da svaki sekundarni indeks za hvata uvek samo jedno polje. Ovim indeksima se dodeljuju ekstenzije od X01 do XFF zavisno od loga po kojem je polju indeksiranje izvrseno. Postoje dve vrste sekundarn h indekse oni koji se ažuriraju svaki put kada se izvrši neka izmena u tabeli (INCSECONDARY) i oni koji se ne ažuriraju (SECONDARY). Kod INCSECONDARY indekse se svak put manje semo onaj deo datoteke koji zahteva izmenu. Pošto je indeks u svakom trenutku ažuran sve pretiežnije I upiti se odvijaju i razmerno brže nego kod sekundarnog indeksa bez ažuriranja. Kod SECONDARY ndaksa se međutim prilkom prve sledeca izmene u tabeli čitav indeks progiseka va za zastareo i kompleino se regenerise prvi sledeci put kad zatreba

Kompozitni (mapirani) indeks se kreira na osnovu više polja pn čemu nj hov broj i međusobni položaj nisu od značaja. Ovim indeksom vam je omoguceno da indeksirate proizvoljan broj polje kao da su jedno veliko spojeno polje. Datoteke kompoztnih indeksa mayu eks lenzija od XG0 do XGF. Za jedno isto polje ne možete imati i sekundarn i kompozitni indeks (koji zahvata sa mo njega). Uslov za kreiranje INCSECONDARY I kompozitnih indekse je posloj anje primarnog ndeksa. Za SECONDARY indeks lo nije neophodno. Velika dužina datoteka sa sekundarnim indeksime koji čak pretazi i dužinu same tabele jeste ono što bi trebalo da nalara Borlandove programera da se malo zamisle

Kreirenje indeksa

Kreiranje indeksne datoteke se mora obavi ti *pre* otvaranja tabela. Treba primetiti da se indeks dodeljuje tabeli na osnovu njenog naz va jer u tom trenutku tabe- la još nije otvotena i nema svoj ID broj koji bi služio umesto imena. Uopste važi pravilo svaka funkcija koja ima kao parametar ime tabele moze se uspešno izvršiti samo ako je tabela zatvotena.

Kre anje indekse ne zavisi od njegovog tipa uvek se koristi funkcija **PXKeyAdd** čiji prototip izgleda ovako

```
PXCODE PXKeyAdd( char *ImeTabele, int BrPolja,
FIELDHANDLE *IdPolja, int Tip)
```

ImeTabele je kompletna putanje i ime tabela e eksten zija se formira prema tipu indeksa

BrPolja odreduje koliko polja treba da zahvati indeks. Za primarni indeks je to broj polja koja čine ključ a u slučaju sekundarnog ovaj broj treba de bude 1 jer sekundarni indeks može biti kreiran iz samo jednog polja. Kompozitni indeks iako formiran iz više polja predstavlja jedno logičko polje pa se i ze njega ovde stavlja 1

IdPolja je adresa niza u kojem su ID brojevi polje koja treba da udu u indeks. U slučaju primarnih indeksa ovaj podatak se ne koi sti jer je broj polja u indeksu (**BrPolja**) dovoljan podatak za kreiranje – zna se da je reč o početnim poljima u slogu da njihovi ID brojav nisu kontinualno od jed nice. U slučaju sekundarnih indeksa je potreban i dovoljan se mo prvi clan niza jer se indeks i kreira za samo jedno polje. U tu prvi clan se smešta ID broj polja po kojeme se indeks kreira (1-255). Kada je rač o kom pozitnim indeksima onda se u prvi clan n ze smesta ID broj koji vraća funkcija **PXKeyMap** (> 255). Da kle ispada da je napotrabno da se ovoj funkciji proslediye čitav niz kad je dovoljan i samo jedan clan. Ovo je učinjeno zbog logo što se u narednim varzija ma plan raju INCSECONDARY indeksi preko više polja a lada ce bit potreban niz

Tip je bp ndekse koji treba kreirati: PRIMARY SECONDARY ili INCSECONDARY

Pogladajmo kako izgleda kreiranje indeksa u praksi. Kako bez primarnog ndeksa ne vredi ni pomišljati na ostale kreiranjno prvo njega. Na listingu 1 je prikazano kako izgleda kreiranje sekundarnog indekse bez održavanja po trečim polju i sa održavanjem po petom polju

Pom njali smo situacije u kojima vas program treba da radi se labelama čije strukture nisu unapred pozna le i tom prilkom smo naveli da postoji mogucnost da saznate koliko je polja u takvoj labeli pod primarnim indeksom. To vam omogućava funkcija **PXKeyNFlds**. U svakom slučaju bitno je reći da je dovoljno da (spravna) datoteka primernog indekse postoji na disku da bi ta labela bila indekslrana. Ako se greškom obriše PX datoteka ona neće biti automalski kre rana vec to morate obaviti sami (funkcija **PXKeyNFlds** se moze koristi h za detekciju primarnog indeksa)

Kada prestane potreba za nekim indeksom njegovo uklanjanje iz lamlje se obavlja funkcijom **PXKeyDrop**. Ova funkcija ce fizički ukloniti odgovarajucu datoteku sa diska. Ako uklanjate primarni indeks onda ce automalski biti obrisani i svi ostali indeksi. Kao i druge funkcije ova grupe i ova se mora pozvati nad zatvorenom labelom. Kao drugi parametar (prv je ime naravno) prosleduje se ID broj indeksa koji treba ukloniti. Tej parametar je isti onaj koji ste prosledivali funkciji **PXTblOpen** kada ste želeli da se tabela otvori po indeksu koj on predstavlja

Kompozitna/mapirane polja

Veliki nedostatak prethodna verzije bila je nemogućnost da sekundarni indeks zahvati više polja. Sa novom verzijom je ovaj problem rešen stvorene su mapirana polje. Mapirana polje nastaju pridruživanjem jednog li više polja jedinstvenom ID broju. Svaki put kada upotrebite taj ID broj on će telensati na njemu odgovarajucu grupu polja. Primetite da sa polja na ovaj način grupisu isključivo logicki lj nema nikakvog fizickog grupisanja polja u tabeli. O ovoj grupi polja možete ji trebo lo bi da razmisljate kao o jednom jedinom polju nasta lom njihovim zamišljenim spajanjem. Razume se u ove ko napravljeno polje n ma moguce ništa upisati niti iz nje ga procitati ali ono i nije za to namenjeno. Svrha upotrebe ovakvih polja dolazi do izražaja prilikom sortiranja ili pretraživanja kada vam je potrebno da slogove uredite na osnovu vrednosti više polja. Ii kada treba da pronadete slog po nekom složenijem uslovu. Otud se ID broj ovakvog polja može upotrebiti samo u funkcijama koje imaju veze sa indeksiranjem. **PXKeyAdd PXKeyDrop PXTblOpen PXSrchFld** i **PXFldName**. Mapirana polja se mogu koristiti za pravljenje ndeksa neosetljivog na velika i mala slova (*case-insensitive*) pa u tom slučaju postoji opravdanje za mapiranje samo jednog fizičkog polja. Kada formirate mapirano polje ID broj koji za njega dobijete ne bi smeo da se poklopi ni sa jednim od (maksimalno) 255 polje u tabeli. Zato vam funkcija koja visi mapiranja vraća ID broj koji će uvek biti veci od 255

Ime le funkcije je **PXKeyMap**

```
PXCODE PXKeyMap(char *ImeTabele, int BrPolja,
FIELDHANDLE *IdPolja, char *ImePolja, int Mod,
FIELDHANDLE *IdIndeksa)
```

Ovim pozivom ce se labeli **ImeTabele** dodati jedno zamišljeno logičko polje sledecih karakteristika

BrPolja – broj zahvaćenih polja (od 1 do 16)

IdPolja – niz ID brojeva polja koji ga čine. Ii ID brojevi ne smeju odgovarati nijednom postojecem sekundarnom Ili kompozitnom indeksu. nti smeju odgovarati BLOB polju

ImePolja – ma polje koje mu se dodeljuje (svako polje mora imati ime pa i logički),

Mod – 0 – za pravljenje razlike između malih i velikih slova
– 1 nema razlike

IdIndeksa – adresa promenljive u koju se smešta ID broj mapiranog polja

Važna napomena funkcija **PXKeyMap** ne kreira nikakav indeks one samo pamti grupu polja. me koje ih reprezentuje i dajuci vam ID broj za njih dozvoljava da ih nadalja tretirate kao jedno polje. Tek nakon što ste ne ovaj nacin dobili ID broj pozivate funkciju **PXKeyAdd** da biste kreirali odgovarajuci kompoztni indeks. Za ime mapiranog polje važe isti uslovi kao i za obicno polje. ali ako date ime koje vec nosi neko polje u toj tabeli ova funkcije nece ništa uraditi – samo ce vam vratiti ID broj tog polja – o tome vodite računa

Opet kada je potrebno saznati kako izgleda indeks kojim je indeksirana neka nepoznata tabela ne raspalagenju je funkcija **PXKeyQuery** koja daje osnovne podatke o njemu. Informacije koje ce bili vracene su iste ona koje sa prosleduju funkciji **PXKeyMap** pa ih ne trela posebno objasnjivati

Primari mapiranja polja i kreiranja kompozitnog indeksa je del ne list nju 2

Mapirane polja daju novu dimenziju indeksima u *Paradox Engine* biblioteci. Ona pružaju zadovoljavajuca rešenja all ipak se tokom rada može osetiti nedostatak indeksa po nekom zrazu. Postoje određeni problemi vezani za postavljanje „filtera" na neku tabelu jer tako nešto nije moguće uraditi postojećim funkcijama. Nataviono nije nemoguća napraviti tako nešto ali bilo bi mno

go lepše da *Borland* i o tome povede računa u daljem razvoju ove biblioteke

PRETRAŽIVANJE BAZE

Ovde ćemo opisati samo osnovnu tehniku pretraživanja. Dakle postupke koje treba sprovesti da bi se pronašao odredjeni slog u tabeli. Priča za sebe su složenije situacije u kojima treba pronaći i izvršiti sve slogove koji zadovoljavaju neki (proizvoljni) kriterijum pretrage ali njihov prikaz prevaz lazi okvire ove serije članaka.

Pogledajmo šta nudi *Paradox Engine* pretraživanje po primarnom indeksu sekundarnom indeksu kompozitnom indeksu i nein deksiranom polju. Za sve pomenute načine pretraživanja važi da je moguće traziti prvi slog u tabeli koji zadovoljava (SEARCHFIRST) il sledeći slog u odnosu na tekući (SEARCHNEXT). Ne želiš *nije* moguće vršiti pretragu unatrag (nešto kao SEARCHPREV). Prikom pretrage može se uzimati u obzir razlika između velikih i malih slova (to je jedini način kod primarnih i JINCJSECONDARY indeksa) ili se može zanemariti (što je slučaj kod kompozitnih indeksa). Može se tražiti i neki približni sadržaj (CLOSESTRECORD) Više od ovog morate pravili sami.

Princip pretraživanja je sledeći slog baiter (otvoren uz dalju tabelu) popunjava se **IPXPut**.) uzorkom koji želimo da nadjemo Popunjava se samo ono polje (ili polja) po kojem cemo vršiti pretragu Zatim se poziva jedna od funkcija za pretraživanje **PXSrchKey** ili **PXSrchFld** kojoj se naznačava po kojem polju iz slog baiera (ili mapiranom polju) traba da vrsi pretragu Sadržaj ostalih polja u slog baieru neće imati ulicaja na tok pretrage Funkcije **PXSrchKey** se koristi za pretraživanje po poljima zahvaćenim primarnim indeksom a funkcije **PXSrchFld** za pretraživanje po ne ndeksiranim polji ma (ili indeks tanim ako ze zadato polje posloji in deks) Obe funkcije ca vraiti **PXSUCCESS** ako je slog pronaden, i **PXERR RECNOTFOUND** u slučaju da ta kav podatak ne postoji Ako je slog pronaden on posta je tekući To ne znači da je i automatski ucitan u slog bafer Ta operacija morate izvesti sami Zadrzimo se za kratko na pomenulim funkcijama Najpre prototipevi

PXCODE PXSrchKey(TABLEHANDLE idTabela RECORDHANDLE idSlogBafera Int PoPolju Int Mod)

Parametar **idTabele** predstavlja ID broj tabela koji se dobija prilkom otvaranja (**PXTblOpen**) **idSlogBafera** je ID broj sloga u kojem je uzorak koji se traži. Parametar **PePolju** naznačava koliko ce polja koja čine primarni indeks bi uzeto u obzir prilkom pretrage ako (ne pnmer) primarni indeks čine prva tri polja onda je moguće vršiti pretragu samo po prvom polju po prvom i drugom polju ili po sva tri. U zavisnosh od toga para meter **PoPolju** će uzeti vrednosti 1 2 ili 3 respektivno Parametar **Mod** odredjuje koji se slog traži prvi u tabeli koji zadovoljava (SEARCHFIRST) sledeći (SEARCHNEXT) ili prvi približni poemešano od početka tabele (CLOSESTRECORD)

PXCODE PXSrchFld(TABLEHANDLE idTabele, RECORDHANDLE idSlogBafera FIELDHANDLE PePolju Int Mod)

Parametri su isti kao kod funkcije **PXSrchKey** jedino parametar **PoPolju** sadrži ID broj polja po kojem se vrši pretraga Za razliku od funkcije **PXSrchKey** ova funkcija ne zahteva da tabela bude indeksirana da bi se pretraga izvršila. S tim u vezi mogu nastati in situacije

- ako je uz tabelu otvoren primarni indeks, onda je moguće vršiti pretragu po bilo kojem sekundarnom ili kompozitnom indeksu
- ako je uz tabelu otvoren sekundarni indeks onda je moguće pretraživanje samo po tom indeksu
- ako uz tabelu nisu otvoreni indeks vrši se prosto sekvencijalno pretraživanje i tada parametari **Mod** ne može imati vrednost CLOSESTRECORD

Kada treba da koristite indeks kod kojeg se pravi razlika između velikih malih slova *case sensitive*) upotrebite primarni ili sekundarni indeks. Za koriscenje indeksa kod kojeg nema razlike između velikih malih slova (*case insensitive*) upotrebite kompozitni indeks. Sasvim je regularno da u toku programa otvorite više puta jednu istu tabelu svaki put po indeksu koje vam odgovara. Za svako otvaranje cete dobiti novi ID broj. Tako cete za pretraživanje po bilo kojem od indeksa jednostavno upotrebiti njemu odgovarajuci ID broj tabele. Primet ze to je del zavedno sa ostalim pomenima koj pisate ovaj tekst a koji ce se uskoro naci na Sezamu u diskelonjmu R:\RSOFT

(Sezam klisa)

Nastavak sa strane 8

sal posto nešto što je ranije koštalo milion sada košta 300 000 maraka. I to je otvorno stravično kada čovek shvati da oni sade kupuju kupca Cela ideja u trgovini je de prodevac prode nešto e kupac to plati Zamisli kada neko more tebi de plati da bi ti njegovu robu uzeo Automatske se pita „Gde mi je pečurka? Gde mi je hrana? Ko je upalio svetlo? Jao čime su me hranili?" To je grozan osećaj Ja nikada ne govorim kao marketing oni dolaze posle Ja samo kažem činjenice, tehniku

Jednostavno postaju sveesni da će to što treba da kupe 32-bitna mašinua postar bajato već kroz 2 do 3 godine A čovek uopšte ne kupuje ta veće mašine tek tako ne dve tri godine Radnu stanicu još može to se tako zamanjuje to je kao front end mašine eli centralne mašine I infrastruktura ako je 32 bitna je hendikep svoje vrste.

Računari A da li DEC još prodaje mašine zasnovane na K4000?

Pol Krneta- Da, to kupuju ljudi kojim je upkos svemu, potrebne ta mašine, jer imaju okruženje „stenih" dobnih procesora I uvek imaju opciju Alpha Ready DECstation Alpha Ready VAX Kada prode pole godine godine aplikacije je na Alfi i oni vraćaju tu mašinu a uzimaju Alfu sa popustom

Računari Koji ljudi su glavni za razvoj Alfe?

Pol Krneta To su Dick Sites i Richard Witek – oni su samu erhitekturu Instrukcije *op-code* I arhitekturu sa aspekta hardvera definisali Za Implementaciju je kasnije bilo potrebno puno ljudi de se to sve od ideje realizuje u hardveru Posle je bilo na stotina pa l dve hiljade ljudi u jednom trenutku uključeno u projekal Alfa Bukvalno svi resursi su bili uključeni u projekal Alfa Ljudi koji su već lonsko imali debelo iskustvo u razvoju raznoraznih VAX-ove 9000-rki RISC mašine multiprocesora, firme Cray Bile je to velorna kompleksne operacija. Razvijati multiprocesing bez racino Cray e bilo bi vrlo neodgovorno I kratkovido Čovek hoća da prvi mašinu koja je stvarno prva te vrste na svetu ali kompletno de bude zdrava i kroz pet desel godina Rich Witek je bio prešao u Apple i radio je na razvoju PowerPC-ja sada se vratio ponovo u DEC i radi na razvoju implementacije Alfe za *low cost* mašine za commodity market za komunikacije, laserske pnniere video igre Jedna grupa upravo je stvari razvija *high volume Alphas* de budu vrlo jeftine Danas jedan Alfa procesor kada se kupi košta oko 400 dolara To je upola jeftinje od Pentiuma. Mali je brz, ima 1 7 millone tranzistora Tek je u stvari početak

Medutim izdvojiti samo ove dve čoveke, bilo bi nepravedno Oni sami mogli bi de napišu papii sa definicijom mašine i da od toga ne bude ništa de se nije pojavile druge grupa koja pravi čipove i bavi se samom tehnologijom poluprivodnika Danas na tržištu nema mašine koja se komercijalno prodaje i radi ne 150 MHz Čak i 100MHz je problem svime Kod nes as rutinski prave mašine od 200 III 300 MHz koje es prodaju il se vrte po laboratorijame To je lo što je dolo taj faktor performanse Alfi Danas je sval postao toliko lažak toliko oštar de samo takve stvari mogu de produ Jer kome treba još jedan RISC procesor? Toll ko dobnih ima ne ovome svetu

I uvek je smešne situacije – Alfa u računskom centru To stoje neke velike kante ogromne teške po nekoliko lona se onim cevima koje dovode i odvode neke stvan tečne ili se puno struje I onda es pojavi mala kutijica veličine PC-ja Obično as one prvo kupe A možne su i srazmerno im ja niska cena Kao eto n zik kupicemo je pa da vidimo I uvek je situacija paradoksalna Ljudi koji dolaze sa velikih mašina izgube odjednarput orijentaciju šta je veliko a šta je malo Najbolja ilustracija odnosa prema tim mašinama su kućni nazivi Speedy DB-Fast Gonzales I Gonzales II Hurry

ZANIMLJIVOSTI

Virusi

Deseti rođendan kompjuterskih virusa

Iako su „Računari" posvatili dosta prostora kompjuterskim virusima i borbi protiv njih malo ko zna kada su oni tačno nastali i ko im je idejni tvorac Možda je 11 novembar deseh rođendan prvog kompjuterskog virusa dobra prilika da kažemo nekoliko reči o tome

Kompjuterski virusi su ovih dana proslavili deset godine života i „uspešnog" rada 11 novembra 1983 godine (da smo znali pokrenuli bismo Sezam dan ranije) Fred Kohen (*Fred Cohen*) tada student završne godine Univerziteta Južne Kalifornije (USC), prikazao je Leonardu Adlemanu profesoru koji mu je predavao o sigurnosb kompjuterskih sistema prograrn koji napada druge programe i „razmnožava" se bez znanje korisnika Profesor Adleman je, pošto je proučio dejstvo programa, prvi pomenuo ime „kompjuterskivirus"

Autor prvog (doduše „dobroćudnog") virusa Fred Kohen u međuvremenu je doktorirao i sada je šef odeljenja za analitiku upravljanje u Hadsonu Oha jo Objavio je preko 30 članaka i nekoliko knjige koje sa bava problemom virusa Tranutno piše knjigu *It's Alive*" koje es naravno bavi virusima I koju će tokom 1994 godine objaviti izdavačka kuća *John Wiley & Sons*

Prvi njegov rad na tu temu izložen je spnia 1984 godine na IFIP (*International Federation of Information Processing*) konferenciji Što se same ideje tiče kaže da mu je napuisto pala ne pamet dok je jednom „nevino" sedeo na času Takode u intervjuu za elektronski časopis *Newsbytes*, kaže da njegovom profesoru ideja nije delovale naročito privlačno – ko bi još gubio vreme na nešto što je tako jednostavno I tako očigled no Slično su reagovali I ostali naučnici koji su u po četku bili više nego indiferantni prema Kohenovom radu tek kada su virusi stveino počili da predstavlja ju opasnost „prisetili" su se nekih njegovih zaključa ka Profesor Adleman koji je ovakvim programima dao ime kaže da je moguće da je ime korišćeno i ranije u jednom SF romanu Izdatom još 1960 godine Pe ipak 1983 godine je prvi put to ime korišćeno da sa opiše jedan reelan program de je tada kompjuterski virus drugačije nazvan svakako bi se danas konsto tej strugi termin

Profesor Adleman smatra da neme svrhe „kriviti" Freda Kohena za ovaj pronalazak koji je mnoge ozbiljno namučio i nepravio ogromne štete – to je moralo da se desi pra III kasnije a u pitanju je fundamentalno naučno istraživanje što je uostalom i posao onih koji se bave sigurnošću kompjuterskih si stema Zapravo virusi tj ideje na kojima počivaju) sa mogu i korisno primeniti naročito kada se radi o računarskim mrežama Tada lakvi programi mogu da u klanjaju privremene i napotrebne delotleke se diska, proveravaju verzije programe i svime slavljaju na ra spolaganje najnovoju optimizuju sistem i olakšavaju upravljenje mrežom Neki postojeći programi obezbe đuju kompatibilnost es drugim programima (pominju se QEMM i *Windows 3 1*) konsteći mehanizme koje su detaljno razradili bas autori virusa – „prearatanje" program pri učitavanju provere njegovog elektronskog „potpisa" i intervencije na kodu

Ne bi naravno svako „lek leko" pustio jedan virus mekar i dobrođušen i konstan da „šeta" njegovom mrežom ko zna šta bi sa mogilo dogodili i baz namere njegovog autora! Di Kohen ime odgovor i ne lo – praktično svi komercijalni paketi imaju neke bagove i pod nekim okolnostima proizvode čudne rezultate koji mogu dovesti i do prilične šteta Ne primer instalaciona procedura za *Windows 3 1* imala je bag zbog koga kada korisnik naredi brisanje README datoteka koje služu uz *Windows* pobriše i ava ostale README datoteke na disku Da li zbog mogućnosti da postoji još neki lakav bag treba da prestanemo da koristimo *Windows*?

Dejan Ristanović

IBM HEWLETT PACKARD digital EPSON SUN FUJITSU BOCA

U vremenu zastoja...
...nešto zaista NOVO
uskoro
Prezentacija
procesora XXI veka.
DEC Alpha AXP
uvodi nas u svet
ergonomije i
visokih performansi!

Višekorisnički
microVAX-sistemi
od sada po ceni
personalnih
računara!

Za one koji znaju
šta im treba

Projektovanje informacionih sistema, servis i prodaja
I Sutjeska 19a, 11210 Beograd
Tel:711-9 | 6 Tel/Fax:711-478

IBM HEWLETT PACKARD digital EPSON SUN FUJITSU BOCA

Sarađujući sa firmom RIBBON možete uštedeti znatna novčana sredstva, eliminišući kupovinu novih traka za ava tipove štampača i pisaćih mašina Mi u postojeća kućišta, koja predlažemo da sačuvate, ugrađujamo novu traku naograničani broj puta, dok god na dođe do fizičkog oštećenja kasete

Ovim načinom mi amo minimum 30% do 50% jaftiniji za bilo koju traku u odnosu na ave anabdevača RIBBONA. Nova traka koju ugrađujemo traja duže i daja izuzatno jasne otlaka, jar dolazi direktno od amaričkog i nemačkog proizvođača I na zadržava sa u akladištima, te teko ne dolazi do aušenja ribbona

Saredujemo sa mnogim beogradakim firmama kao I sa firmema širom Srbija Uvaravamo Vaa u efikaanost i kvalitet našeg rada I nadamo sa da ćemo ostvariti saradnju na obostrano zadovoljstvo.

OTISAK KOJI NEMA PREMCA

Šta je Clean Boot za Windows

Grubo parafrazirano latinska i filozofija nas uča da je religija nastala delom i zbog toga što čovek nije mogao racionalno da tumači neke prirodne i društvene fenomene. Problemi sa instalacijom *Windows*-a i nerazumevanje šta, zapravo znači jedan tako složen sistem grafičkog interfejsa (ili operativnog sistema proizvode novu vrstu vernike, one koji *Windows* nazivaju *Widows* (Udovica) *Pandžeri* i slično, koji ne znaju da ga instaliraju i konfigurišu ili da upotrebimo eufemizam imaju problema sa *Windows*-om. Posebna kategorija, koja već zadire u sferu klasnih odnosa, su oni čiji hardver nije dorastao zahtevima *Windows*-a, pa ih muče i posebna muka.

Neka od muka mogu se rešiti ili im se bar može naći uzrok ako koristimo ono što Microsoft zove *Clean Boot*: pokretanje računara nakon što se iz CONFIG.SYS i AUTOEXEC.BAT datoteka uklone sve komande koje nisu neophodne. To je prvi korak u otklanjanju problema koji nastaju zbog konflikta *Windowse*-a i nekih rezidentnih programa i drajvera za različite delove sistema. Postoji više različitih *Clean Boot* procedura, koju treba koristiti zavisi od toga da li je *Windows* već instaliran ili ne.

Prvo čete morati da kreirate disketu sa koje ćete podići sistem. Disketa treba da sadrži sve COS sistemske datoteke i programe koji su otrebni da bi sistem bio potpuno funkcionalan (programi za kompresiju diska, za particioniranje diska, drajveri sa eventualne SCSI diskove i slično)

Clean Boot pre instaliranja

Ako ne koristite softver za particioniranje diska (kao što je *Disk Manager* firme OnTrack) kompresiju diska (Stacker, Double Space i slično) i druga drajvera za specijalne komponente sistema, računar sa može pokrenuti sa diskete koja sadrži samo sistemske datoteke MS DOS-a bez CONFIG.SYS i AUTOEXEC.BAT datoteke. Ako sa pomenuti programi ipak koriste da toteke koje se koriste pri startovanju sistema bi trebalo ovako da izgledaju

```
CONFIG.SYS
FILES=45
BUFFERS=20
DEVICE=C:\WINDOWS\HIMEM.SYS
<program za particioniranje diska>
<program sa kompresiju sadržaja diska>
<ostali neophodni drajveri>
STACKS=9,256
SHELL=C:\<punl path>COMMAND.COM /E:1024 /P
AUTOEXEC.BAT
PROMPT $P$G
PATH=C:\WINDOWS;C:\DOS;C:\
SET TEMP=C:\<važeća staza do TEMP direktorijuma>
```

Izuzeci

Postoje drajveri koji se ne smeju uklanjati iz konfiguracionih datoteka operativnog sistema. Lista nije potpuna, ali sadrži najveći broj drajvera koji su u čestoj upotrebi

Drajveri za tvrde diskove:

SQY55.SYS, SSTBIO.SYS, SSTDRIVE.SYS, AH1544.SYS, *ILIM386.SYS*, ASPI4DOS.SYS, SCSIHA.SYS, SCSIDSK.SYS, SKYDRVI.SYS, AT_DOSXL.SYS, NONSTD.SYS

Softver za particioniranje diska:

DMDRVR.BIN, SSTOR.SYS, HARDRIVE.SYS, EVDR.SYS, FIXT_DRV.SYS, LDRIVE.SYS, ENDISK.SYS

Programi sa kompresiju sadržaja diska:

STACKER.COM, SSWAP.COM, SSTOR.EXE, DEVSWAP.COM

Ako u konfiguracionim datotekama otkrijete program ili drajver za koji ne znate šta zapravo radi najbolje je da ga ostavite na miru. Možda ćete moći i sa njim da pokrenete sistem. Pogledajte šta se izpisuje tokom startovanja, jer sa većina drajvera i ostalog softvera oglašava bar nekim porukom.

Clean Boot ne obuhvata sledeći softver i komande

- DOS=HIGN,UMB
- EMM386.EXE
- INSTALL=SHARE.EXE
- INSTALL=FASTOPEN.EXE
- razne spoljne programe za upravljanje memorijom
- RAM disk softver
- JOIN, GRAPHICS, PRINT, SUBST, APPEND
- MCOE komandu za preusmeravanje štampanja
- višestruke PATH naredbe
- drajvera za miša na nivou MS DOS-a
- razne spoljne programe za disk cache
- razna rezidentne (TSR) programa
- LOGIMENU i CLICK. Logitech-ove programe za miševe
- programe sa detektovanje prisustva virusa
- drajvera sa skenera i fax uređaje
- drajvera za CD ROM uređaje i mrežne uređaje
- softver za backup na traku preusmeravanje i bafere
- drajvera za uređaje za akviziciju podataka
- programčiće za ubrzavanje tastature

Kada probati Clean Boot

Clean Boot je samo jedan od koraka koji u standardnoj proceduri za traženja uzroka problema preporučuju stručnjaci za podršku kod Microsoft-a. Uzmite u obzir da vaš problem može imati veza sa 386 Enhanced načinom rada *Windows*-a o čemu sam pisao u broju 96 dobro je pogledati da li neki od problema ima veze sa time

Procedura može izgledati ovako

- instalirajte standardni VGA drajver za grafičku karticu drajver se isporučuje ne originalnim diskelama za instalaciju. Koristeći se uvek prvo instalirati *Windows* sa ovim drajverom jer ća na taj način odgovarajuće datoteke biti na disku i kasnije, kada eventualno promenite aktivni drajver
- Pnalmenujte STARTUP.GRP datoteku u njoj se nalaze reference na programa koji će biti pokrenuti od mah po startu *Windows*-a. I koji mogu biti uzrok problema. Označite znakom za komentar (;*) linije LOAD= i RUN= u WIN.INI datoteci. U redovi iskoda sadrže imena programa koji se pokreću po startovanju *Windows* a
- Pokrenite sada *Windows* u standardnom modu komandom WIN /S. Ako sva buda u redu moguće je da je problem vezan samo za Enhnaced način rada
- Ako je problem vezan za Enhanced mod pokrenite *Windows* koristeći komandu WIN /D:XVS (vidi WIN.INI u broju 96 za značenje ovih opcija)
- Ako problem postoji i u standardnom načinu rada probajte *Clean Boot*

Bez spoljnih drajvera...

Jedan od mogućih razloga za peripetije sa *Windows*-om može biti beskonačno nesistematsko čačkanje po parametrima u konfiguracionim datotekama, posebno u datoteci SYSTEM.INI. Ova datoteka sadrži informacije od čije korektnosti zavisi da li će *Windows* uopšte da startuje. Zbog ovoga ćete možda u nekom trenutku morati da rekonstruišete originalni sadržaj datoteke, ili bar onaj koji je najbližnje optimalnim parametrima za vašu mašinu. Osim u situacijama kada otklanjate probleme, "čista" SYSTEM.INI datoteka će vam zatrebati i ako je sadržaj datoteke oštećen ili obrisan

Obavezno prvo sačuvajte početno stanje SYSTEM.INI ako ne trebaloi. Nađite svim originalne distribucione diskete *Windows*-a i ekspandujte datoteku SYSTEM.SR_ sledećom komandom.

```
c:\windows\expand e:\system.sr_ c:\windows\system.ini
```

Ova datoteka se nalazi na prvoj disketi iz seta for mata 3.5" odnosno drugoj iz seta 5.25" disketa. Kod *Windows for Workgroups* datoteka je na trećoj disketi u oba formata.

Predite u *Windows* direktorijum i pokrenite *Windows* SETUP program iz DOS-a. U maski koja pokazuje hardverska opcije sistema sva polja su u početku prazna. Izaberite opcije koje odgovaraju vašem sistemu i nađite iz SETUP-a.

Otvorite SYSTEM.INI datoteku iz nekog ASCII tekst editora i izvršite sledeće izmene

- dodajte PROGMAN.EXE u liniju **SHELL=** u sekciji **[Boot]**
- uklonite liniju **TASKMAN=** iz **[Boot]** sekcije
- ako želite da omogućite 32-Bit Disk Access, dodajte sledeće linije u **[386Enh]** sekciju

```
32BitDiskAccess=<logička vrednost>
device=*int13
device=*wdctrl <logička vrednost> je True ako želite da uključite ovu opciju.
```

- Pokrenite *Windows*

Ako sada ne prijavi nikakve greške uspešno ste kreirali SYSTEM.INI datoteku. Ako dobijate greške S020 ili S021 datoteka koju ste kreirali je možda neispravna. Proverite

- proverite da li postoji više kopija SYSTEM.INI datoteke na različitim mestima na disku
- proverite da li postoje izgubljeni delovi prostora na disku naredbom CHKDSK /F

Graške parnosti u Windows-u

Greške parnosti nastaju nakon generisanja NMI prekida (*NMI interrupts*). Ispravna instalacija *Windows* a ne može generisati NMI prekid – tako bar kaže Microsoft.

Parnost je vrlo bazična provera ispravnosti podataka uz svaki bajt koisnih informacija (8 bita) čuva se i bit parnosti koji se generiše na osnovu broja jedinica i nula u bajtu. Procesor proverava ispravnost bita parnosti i generiše NMI prekid ako nađe na nekonsistentno postavljan bit parnosti. U Enhanced načinu rada postoji virtuelni drajver *parity koji presreće indikaciju NMI i generiše poruku o grešci. U standardnom načinu rada NMI se ignoriše i prepušta se obradu ruini sa obradu prekida parnosti. Rutina ne radi ništa. Ili generiše *fullscreen* poruku o grešci u parnosti to je jedan od čuvenih plavih ekrana *Windows* a, koji obično označavaju grešku od koje se sistem ne može oporaviti

Jedan od načina da eliminišete grešku parnosti je uklanjanje drajvera za neki od uređaja ili reinstaliranog programa. Ovim se menja samo lokacija u memoriji sa koje se izvršava program kod programe ne nalazi se više u zoni neispravne memorije i više se ne generiše upozorenje o grešci. Ako u problematičnom području memorije *Windows* čuva podatke može se desiti da se prekid parnosti uopšte ne generiše. Doduše greške se može pojaviti kasnije jer se kod i podaci smeštaju na promenljive pozicije u memoriji

Windows odvaja memoriju tako da se ona obično inicijalizuje da početni sadržaj bude nula. Zbog toga će bit parnosti biti nula. Ako postoji problematičan čip koji bit parnosti stalno drži na vrednosti nula, sve funkcionlše normalno pošto se ne tu lokaciju nikta ne upisuje bit parnosti je pravilno korektan. Ako sumnjiva adresa upadne u segment koda programa (gde se pohranjuje izvršni kod) vrlo je verovatno de će bi generisan prekid zbog parnosti. Uteratko Microsoft smatra da je *Parity Error* golovo uvek hardverski problem

Arhiviranje i komprasija

Kada biste nekog iskusnijeg korisnika MS-DOS-a pitali da vam preporuči arhiver, odgovor bi verovatno bio *Zip*, *Arc* ili neki drugi popularni program. Za Unix biste dobili odgovor **tar** i **compress**, ili **tar** i **gzip**. Dakle, uvek se preporučuju dva programa. Zašto?

Pogledajmo šta zapravo rade *Zip*, *Arc* i slični arhiveri. Oni, u stvari obavljuju dve važne posla. Prvi je „sabijanje" pojedinačnih fajlova poznatim tehnikama kompresije, o kojima su „Računari" u više navrata pisali, a drugi je arhiviranje, kojim se kompresovani fajlovi sklapaju u celinu, u novi fajl. Ne Unix-u tek odnedavno postoje *Zip*, *Arc* i brojni drugi arhiveri. Mnogo pre njih su nastali programi **tar** i **compress**. Prvi se koristi za arhivliranje, dok je drugi namenjen kompresiji. Do danas se pojavilo mnogo klonova ova dva programa i veliki broj njih se i koristi, ali ono što je ostalo jeste koncept odvojenog arhiviranja i kompresovanja.

Najčešće korišćeni program za arhiviranje je stari ali dobro držeći *tar*. Primera radi da biste arhivirali tekući direktorijum i sve poddirektorijume u datoteku *arhiva.tar*, dovoljno je da otkucate

```
tar cf arhiva tar *
```

Da biste dearhivirali ovu arhivu, otkucajte

```
tar xf arhiva tar
```

Osim opcije „c" (*create*) i „x" (*extract*), moguće je dodati nove fajlove u arhivu (opcija „r"), pregledati sadržaj arhive (opcija „t") i još mnogo toga. Listu i objašnjenje opcija potražite u help-u za komandu **tar** (**man tar**)

Setimo se sada priče o *special files* iz **/dev** direktorijuma koji predstavljaju odgovarajuće uređaje. Kao što smo već napomenuli i tu fajlove možemo posmatrati kao i sve druge. Otuda logično prizlazi da se, umesto u fajl arhiviranje može obaviti i direkino na nekom od uređaja. Tako, ako flopiju odgovara *special file* **/dev/rfd0135ds18**, tada se arhiviranje na ovaj uređaj obavlja sa

```
tar cf /dev/rfd0135ds18 *
```

Ako želite malo da se poigrate, arhivu možete izlistati i na svom terminalu se

```
tar cf /dev/tty
```

Arhiviranje na uređaj uz pomoć **tar**-a je standardni sistem za bekap podataka na Unix-u. Naravno, **tar** podržava smeštanje arhive na više volumena (disketa, strimer traka...).

Uz to, da ne biste morali da razmišljate o komplikovanim nazivima specijalnih datoteka i da biste skratili kucanje, na nekim Unix-ima (SCO) su u **/etc/default/tar** datoteci definisane oznake za najčešće korišćene uređaje (disketne jedinice, strimer jedinice i slično). Tako, umesto nezgrapnog

```
tar cf /dev/rfd0135ds18 *
```

dovoljno je

```
tar c0 *
```

ako je sa „0" definisan **special file** **/dev/rfd0135ds18** u **/etc/default/tar** datoteci. Slično (radi na svim Unix-ima), umesto

```
tar cf /dev/tty *
```

kojim prikazujemo arhivu na terminalu, jednostavnije je otkucati

```
tar cf - *
```

čime se izlaz preusmerava na **stdout** (*Standard Output*). Analogno, sa

```
tar xf
```

kao ulaz koristi se **stdin** (*Standard Input*). Preusmeravanje na **stdout** i sa **stdin** se može korisno upotrebiti na brojne načine, a evo jednog zanimljivog. Ako želite da iskopirate sve fajlove tekućeg i svih poddirektorijuma zajedno sa strukturom direktorijuma u neki drugi dir, probajte sledeće

```
cd pelazni_dir; tar cf -  | ( cd edra
disni_dir; tar xf - )
```

Dobijanu arhivu kao i bilo koji drugi pojedinačni fajl, možete i kompresovati jednim od programa za kompresiju. Najpopularniji su, pre svega, **compress** i nešto manje **pack**, dok se u novije vreme sve češće koristi *GNU Zip* (GZip) koji je kompatibilan sa **compress**-om, ali porad kompatibilnosti i odlje i brže kompresije pruža i brojna druga pogodnosti. Ovi programi vrše kompresiju nad samo jednim fajlom. Dakle, arhiver (**tar**) sjedinjuje više fajlova u jednu arhivu, dok programi za kompresuju (**pack, compress, gzip**) tu arhivu „sabiju"

Arhivu *arhiva.tar* ćemo kompresovati uz pomoć **compress**-a jednostavnim

```
compress arhiva tar
```

i dobiti fajl *arhiva.tar.Z*. U zavisnosti od korišćenog arhivera i **compress** programa, najčešće viđene ekstenzije kompresovanih arhiva su

```
tar z (arhiver tar i kompresor pack),
tar Z (arhiver tar i kompreser com
press ili starije verzija gzip-a),
tar gz (arhiver tar i kompresor gzip
(novije verzija)).
```

Često ćete naći i skraćene verzije ekstenzija kao što su **tz** ili **tZ**, na primer.

Dakompresija je, takođe, jednostavna. *.Z* fajlove, pakovane **compress**-om, respakuje program **uncompress**, *.z* fajlove pakovane **pack**-om, program **unpack**, dok *.Z* i *.gz* fajlove pakovane **compress**-om ili **gzip**-om otpakujte **se gzip -d**

Tekući dir i svi poddirovi

Pominjući u prethodnim redovima fajlove u tekućem direktorijumu i svim poddirektorijuma, sebo sam se koliko mi se samo puta desilo da treba da promenim vlasnika svih fajlova ili, još češće, dozvole za fajlove, u tekućem i svim poddirovima. Po Marfiju, verovatnoća da se to desi je direktno proporcionalna broju poddirektorijuma, tako de me od svake pomisli da idem iz direktorijuma u direktorijum i menjam šta je potrebno, zaboli glava. Kada sam prvi put naišao na ovaj problem, reagovao sam programerski – seo sam i napisao kratak program u C-u koji rekurzivno prolazi kroz direktorijume i izvršava zadatu komandu. Ipak, i tada sam bio siguran da postoji brži i lakši način i bio sam u pravu. Mnogo je jednostavnije napisati *kratak shell script* koji bi eko je potrebno de vlasnik svih fajlova bude korisnik „bin", u najjednostavnijem obliku izgledao ovako

```
#! /bin/sh
tat()
{
    for fnasa in *
    do
        chown bin $fnams
        if tast -d $fnasa
        than
            cd $fname
            tat
            cd ..
        fi
    dona
}
tat
# kraj
```

Kasnije sam našao da *utility* pod imenom **find** može isti posao de obavi na jednostavan i brz način u samo jednoj liniji. Dovoljno je otkucati

```
find    name „*"  exec chown bin {} \;
```

ipak, **find** je pogodan kada na grupi fajlova treba izvršiti jednu ili manji broj komandi. Za ole komplikovanije algoritme mnogo bolje rešenje su *shell script*-ovi

Sa MS-DOS-a na Unix

Pre izvesnog vremena mi je stigao *mail* od jednog korisnika koji je, kao i mnogi pre njega, tvrdio da računer koji administriram ima duhove i tražio da ih hitno isteram. A problem je bio sledeći

Korisnik se do sada susretao uglavnom sa MS-DOS-om i koristio *Borlandov C*. U želji da vidi „kako sve to radi" na Unix-u napisao je čuveni *"hello world"* program i nazvao ga proba.c. Kompajliranje je prošlo bez i jedne greške te je, sav srećan otkucao „proba" – kad, umesto očekivanog „hello world", na ekranu se pojavilo

```
proba  command not faund
```

Objasnio sam mu de je problem u sledećem. Najveći deo *Unix C* kompajlera (da ne budem isključiv, iako nisam video ni jedan koji radi drugačije) kao izlaz daje izvršni program pod imenom **a out** ako mu se drugačije ne „naredi". Dakle trebalo je da program startuje sa **a.out** ili da eksplicitno navede ime izlaznog fajla opcijom **-o**

```
cc  o proba proba c-
```

Kada sam se sledeći put javio na sistem zatekao ma je *mail* u kome je isti korisnik pisao de je otkno nove „duhove". Poslušao je, navodno moj savet i eksplicitno naveo de mu se izvršni fajl zove proba, ali kad otkuca „proba" opet dobije istu poruku, tj. „proba. comamnd not found". Onde sam mu lepo objasnio da po *default*-u, tekući direktorijum nije stavljen u *path* i da mora eksplicitno da navede da se fajl nalazi u tekućem direktorijumu, tj. de otkuca **./proba** ili de tekući direktorijum stavi u *path* u svom *profile* ili *login* fajlu

Nije prošlo ni par dana, sto nove poruke i novih „duhova." Kaže čovek, napravio je program za sortiranje niza brojeva. Kada ga startuje, ne ispisuje mu ništa nego samo stane. Isti program je proveo na svom MS-DOS *Borland C* kompajleru i tamo savršeno radi. Otišao sam do njegovog direktorijuma kad – imam šta da vidim. Čovek je nazvao svoj program *sort.c* i naravno, izvršnom programu deo ime **sort**. Kada sam mu rekao de ubaci svoj direktorijum u *path* on ga je dodao na sam kraj, pa je tako umesto de pokrene program „sort" iz tekućeg dir-a, pokrenuo „sort" iz **/bin** dir-a koji se u *path*-u nalazi pre tekućeg. Ako nije siguran da li takav program već postoji u path-u, treba da otkuca.

```
type <filename>,

ili

whereis <filename>
```

zavisno od verzije Unix-a. Kao rezultat će biti ispisan direktorijum u kome se navedeni fajl nalazi ili poruka de takav fajl ne postoji u path-u

Prekrivanje funkcija

Da bi prevazišli DOS memorijsku barijeru, mnogi komercijalni programi sa pozamašnom količinom koda odavno koriste tehniku poznatu pod nazivom *overlay* – prekrivanje. Mehanizam se zasniva na odvajanju memorijske zone u koju se, po potrebi, sa diska učitava deo programa ukoliko se on trenutno ne nalazi u memoriji. Kada je potrebna neka funkcija koja se u njemu ne nalazi onda sa ista memorijska zona preklapa novim segmentnom koda se diska u kome se nalazi tražena funkcija.

Prva implementacija *overlay* tehnike su bile prilično loše, i bile su veoma neefikasne ukoliko u se u pojedine segmente nisu upisivali praktično samodovoljni delovi programa – čest poziv funkcija između pojedinih *overlay* segmenata je mogao da učini program neupotrebljivim. Jedina prednost ovakvog sistema u odnosu na pisanje više manjih i odvojenih programa je što je pristup zajedničkim podacima polpuno automatizovan – programer o tome uopšte na mora da vodi računa, jer svaki segment programa normalno vidi globalne promenjive celog programa.

Poučen prvim lošim iskustvima, dugo nisam obraćao pažnju na poboljšanja koja su u međuvremenu uveli Borland i Microsoft. *Overlay* tehnika su počela sve ozbiljnije da se primenjuju, naročito kod Clipper-a, kod koga su problemi se količinom koda i DOS memorijom posebno izraženi. Više iz radoznalosti nego iz stvarne potrebe skoro sam pokušao da jeden svoj veći projekat prevedem tako da koristi *overlay* mehanizam i moram priznati da sam prijetno iznenađen. Novi MOVE sistem koji je Microsoft uveo u verziji C 7.00, a spakovan je i u C 8.0 (iz paketa MSVC 1.0) radi zaista dobro i pruža mogućnost da i veliki program uz nešto malo pažljiviju organizaciju ima dosta slobodne DOS memorije u toku svog rada. Verujem da je i Borlandova verzija (pod nazivom VROOM ako se ne varam) podjednako efikasna.

Šta je potrebno učiniti da bi se kreirao program sastavljen iz *overlay*-a? Vrlo malo! Pretpostavimo da se program sastoji iz module numerisanih od 1 do 7 i da se u modulu 1 nalaze funkcije koje koristi najveći broj modula dok ostali moduli predstavljaju manje-više zaokružene funkcionalne celine. Prevođenjem programa u large ili medium modelu, kompajler će kod svakog od modula smestiti u odvojen segment (_MODUL1_TEXT, _MODUL2_TEXT itd) Tačne nazive segmena možemo lako saznati eko pri jednom prolazu ljobičnog" kompajliranja zatražimo od linkera da generiše MAP datoteku

Sada treba kreirati, recimo, PROJ.DEF datoteku sa sledećim sadržajem

```
SEGMENTS
  _MODUL2_TEXT OVL:1
  _MODUL3_TEXT OVL:2
  _MODUL4_TEXT OVL:3
```

i pri povezivanju navesti linkeru ovaj naziv i to je sve. Program će zauzimati mnogo manje memorije stalno će biti aktivan samo _MODUL1_TEXT segment, koji zato što nije eksplicitno alociran spada u takozvani koren (*root*) programa (stalno prisutan segment) a ugrađeni kod će voditi računa da se u memoriji nađe uvek onaj segment u kome je funkcija koja se trenutno izvodi.

Ovde smo kao pojedine *overlay* jedinice koristili segmente ali možemo otići i korak dalje u kontrolisanju overlay-a. Prevođenje programa u formi takozvanih pakovanih funkcija (prekidač /Gy), dobijamo mogućnost da kao *overlay* jedinicu koristimo funkcije. Tada se DEF datoteke sastavljaju u formi

```
FUNCTIONS 1
  _funkcija1
  _funkcija2

FUNCTIONS 2
  _funkcija3
  ...
```

pri čemu broj iza FUNCTIONS deklaracije odreduje u koji *overlay* smeštamo funkcije koje sledе u listi

Postoji određena verovatnoća da će vaš program povezan na ovaj način savršeno raditi i da, sem ogromnih dobitaka u slobodnoj memoriji nećete primetiti nikakve druge razlike. Ipak, verovatnoća je relativno mala jer grupisanje segmenata ili funkcija u *overlay*-eve može presudno da utiče ne ukupno zauzeće memorije i brzinu izvršavanja programa.

Overlay mehanizam koristi keš sistem za segmente – ukoliko je funkcija već u memoriji, nema potrebe da se njen segment ponovo učitava se diska. MOVE konkretno koristi keš čija je veličina jednak zbiru veličina tri najveća *overlay* segmenta. Ako je broj segmenata u programu mali skoro nikakvih dobitaka u zauzeću memorije neće biti jer je za sam keš potrebna memorija. Ako se pak segmenti suviše ustine, neminovno će doći do čestog pozivanja funkcija izmedu više segmenata pa će keš sistem postati neefikasan i program znatno usporiti rad

Sve u svemu, potrebno je određeno planiranje rasporeda funkcija/segmanta i sem pravila da u *overlay* treba smestiti što zaokruženiju funkcionalnu celinu nema drugih opštih pravila. Zato dobra optimizacija može da potraje, all nije komplikovana jer je obezbeđen alat kojim se može pratiti. Alat se kod Microsoft-a zove TRACE. Program se prevede kao da će biti konačan dibager i navede se dodatna biblioteka MOVETR.LIB uz prekidač /NOE kako bi se izbeglo povezivanje ugrađenih poziva iz MOVE.LIB biblioteke. Program zatim startujete i izvedete neku tipičnu operaciju za koju želite da proverite da li je brzina izvršavanja ugrožena novom organizacijom *overlay*-a

Po izlasku iz programe na disku će vas sačekati datoteka MOVE.TRC u kojoj su informacije koje možete pregledati startujući TRACE

```
TRACE /exe program.exe
```

Iz ispisa koji sledi lako se mogu ustanoviti uska grla – za svaki poziv funkcije će pisati da li je već prisutna u memoriji, ili je bilo potrebno učitati segment se diska, koji su segmenti zbog prostora u kešu izbačeni itd. Pošto zbog velikog broja poziva analiza NOVE.TRC datoteke može da bude komplikovana, ona se može ograničiti samo na određeni deo koda. Biblioteke MOVETR.LIB sadrži i funkcije **_movetraceon(void)** i **_movetraceoff(void)**, koja uključu ju/isključuju praćenje reda overlay menadžera

MOVE se može učiniti i efikasnijim ukoliko računar poseduje EMS ili XMS memoriju – umesto učitavanja se diska, *overlay* se može prebaciti u ovu memoriju i potom iz nje uzimati po potrebi. Što može nešto da ubrza rad. Ipak, treba imati na umu da, čak i se ovom mogućnošću loše planiran raspored *overlay*-a na kraju uvek daje spor program

Ukoliko ste pri instalaciji MSC-a instalirali i izvorni kod statup modula među direktorijuma ćete pronaći i MOVE u kome su datoteke MOVEINIT.C i MOVEAPI.H. Prva sadrži kod za inicijalizaciju *overlay* menadžera, i u njoj se može izvršiti intervencija u smislu optimizacije količine memorije koja će biti odvojena za keš. Tako se, recimo može lako izmeniti inicijalno odvajanje prostora u dužini tri najveća *overlay*-a, ili modul prevesti u varijanti koja omogućava da se podešavanje parametara izvodi po potrebi *environ* promenjivima (MOVE_HEAP, MOVE_EMS itd)

Pri upotrebi programa sastavljenog iz *overlay*-a treba imati na umu da je izvršna datoteka stalno otvorena zbog potrebe da se iz nje povremeno čitaju delovi koda. DOS ovu datoteku otvara sa ekskluzivnim pravom pristupa što znači da nećete moći, recimo pod *Windows*-ima, da pokrenete dve kopije programe u dva odvojene DOS taska. U mrežnom okruženju se ovaj problem može rešiti, najčešće davanjem „shareable" atributa izvršnoj datoteci

Osnovno pitanje koje *overlay* tehnika postavlja je pitanje efikasnosti poziva funkcije koje se nalaze u nekom od *overlay*-a. Ako keš rešava pitanje sporosti učitavanje sa diska, što bi zaista predstavljalo problem koliki višak posle po pozivu funkcije ostaje kada se kod funkcije već našao u memoriji? Kako uopšte *overlay* menadžer radi?

Odgovor na ovo pitanje, naravno, zavisi od konkretne implementacije, all se i iz Microsoft ove verzije MOVE mogu izvući neki opšti zaključci. Poziv neke funkcije koje se nalazi u *overlay*-u linker zamenjuje pozivom kratke međufunkcije koja je smeštena u poseban segment. U njoj se uglavnom nalazi poziv centralne tačke *overlay* menadžera koji se izvodi upotrebom INT instrukcije (MOVE koristi Interrapt 3F). Menadžer na stek privremeno smešta sve registre kako njegov rad ne bi poremetio tok programe (pranete parametre vrednost segmentnih registara itd) i zatim proverava da li se konkretne funckija nalazi u memoriji ili ne. Nakon što je po potrebi učita sa diska (uz rešavanje svih referenci – dešava se isto što i inače kada se program učitava u memoriju računara), vraća se steke stanje svih registara i poziva funkciju. Povratak iz funkcije se takoda ne izvodi direktno na mesto poziva, već opet preko *overlay* menadžera – moguće je da se mesto poziva takođe nalazi u nekom od *overlay* a, koji je u toku reda pozvane funkcije moreo biti izbačen iz keša, pa menadžer mora i ovu situaciju da razreši

Sve u svemu, višak posla je prilčan. Konkretne posledice je da će probleme se brzinom izvršavanje imati svaki program koji iz jednog *overlay*-a pozive u neku petlji funkciju iz drugog bez obzira što će se oba segmenta već posle prvog poziva naći u memoriji i tu ostati do kraja izvršavanja petlje. Zato ubrzavanje rada konačnjem EMS ili XMS memorije neće ni malo pomoći programu ukoliko organizacijom nije predviđeno potpuno izbegavanje međusegmentnih poziva u svim makar i malo vremenski kritičnim operacijama. Ne *overlay*-e zato treba i dalje gledati kao na, sa novim varijama već veoma efikasnu, zamenu za varijantu pisanja programa iz više odvojenih modula, a ne kao neko efikasno rešenje za generalni problem DOS memorijskih barijera. Ali na takve situacije smo već prilično navikli – ništa neće biti efikasno kada ispravlje nedostatak iz same srži operativnog sistema.

751 [OS MSDOS 751; PC]: Upgrade sa DOS-a 6.0 na 6.2 je besplatan ako ga preuzmete sa nekog BBS-a, ili košta 10 dolara ako kupite diskete. Oni koji imaju neku od ranijih verzija (počevši od 2.11) treba da spreme 78 dolara – dobićete pun paket koji uključuje i specijalno izdanje knjige Dana Gukina (*Dan Gookin*) zvane nemojte se izvređati! *DOS For Dummies*

Naše čitaoce će „modernaši upgrada" svakako najviše zanimati. Dovoljno je da se Sezame preuzmete program STEPUP koji će potpuno automatski prepraviti vaš DOS 6.0 u 6.2 – jedine nevolje je što prenos „malog dodatka" može i te kako da potraje, jer je originalne datoteke duge čak 1.4 megabajta pa je morala da bude podeljena na više delova.

Napomenimo da za uspešan *upgrade* (proveren!) morate imati komercijalnu verziju MS DOS-a koju ćete prepoznati po datumu na uslužnim fajlovima – ako vidite 10. mart 1993. sve je u redu. Ako, međutim imate verziju od 22. februara 1993. STEPUP će se ljubazno zacuditi kako to da kao jedan od *Microsoft*-ovih beta testera, niste dobili konačnu verziju DOS-a i daće vam telefon (u USA) na koji se možete obratiti da biste je dobili [preuzeto se Sezama, prilog **Zoran Milosavljević** (*zomi*)]

752 [OS.MSDOS; PC]. Evo i nečega za one koji nisu prešli na „deluxe" DBLSPACE već i dalje preferiraju *Stacker 3.0*. Pretpostavlja se da uz njega koristite i QEMM – dodajte QEMM liniji CONFIG.SYS-a jedno /DBF=2 a *Stacker* liniji /EMS i uštedećete 16 kilobajta gornje memorije, naravno na račun EMS-a. Ako ne koristite *Windows 3.x* možete da uštedite još kilobajt osnovne memorije dodajući QEMM liniji prekidač NW3 [preuzato se Sezama, prilog **Kristijan Lazić** (*kriss*)]

753 [SIS, PC]: U raznim knjigama i *help*-ovima pročitaćete da *Windows* ne podržava prozore veće od ekrana. Tvrdnje, međutim nije tačne – pozovite funkciju *MessageBox()* sa desetak kilobajta teksta i videćete (na primer iz *WinSight*-a) da je prozor veliki čitavih 500x8000 tačaka, što je definitivno veće od rezolucije većine ekrana. Jedino je pod takvim uslovima pomeranje vidljivog dela prozora nešto otežano

Kad smo već kod veličine prozora, jeste li znali da sa može menjati i dimenzije desktopa? Poku šajte sa *MoveWindow (GetDesktopWindow(), 0, 0, 10000, 10000, TRUE)* – poziv je nategalan i kao ta kav zavisi od grešičkog drajvera ali radi! [preuzeto sa Sezama, prilog **Obrad Bjelić** (*obj*)]

754 [SIS; PC]: EISA računara je i kod nas sve više, pa neke (naročito sistemske) programe treba prilagođavati ovoj magistrali. Prvi korak je detektovati EISA PC, u čemu može da pomogne program se slike 1. Pošto ste detektovali EISA računar pozovite drugi potprogram sa iste slike i odradite njegov tačan tip [preuzeto se BIX-a, prilog scosta]

755 [KOMS; PC]: Smešan i gotovo simpatičan beg Turbo Linkera u režimu EXETYPE WINDOWS ako se u DEF datoteci u okviru imate module na da karakter 255 (Alt 255). TLINK će tako temeljno „zaglupeti" računar da će ga samo *hard reset* (ili off/on) povratiti [preuzeto se Sezama, prilog **Obrad Bjelić** (*obj*)]

756 [OS.WINDOWS; PC]: *Windows* ažurire grupe *Program Manager*-a (GRP datoteke) tek po završetku rada – zato će, ako nešto „pukne" izmene biti izgubljene i moraćete da ponavljate posao. Trik: držite pritisnut Shift i aktivirajte **File** – *Exit* iz *Program Manager*-a. GRP fajlovi će biti ažurirani a *Windows* neće napuštati [preuzato se Sezama, prilog **Zoran Kohler** (*zkohler*)]

757 [OS.4DOS, PC]: Ako imate probleme sa neredom EXCEPT 4DOS-a proverite da li komanda ne koju je primenjujete ignoriše atribute datoteka. Naime, ova komanda radi tako što prvo setuje *hidden* atribut svim datotekama ne koje ne treba primeniti komandu, pa tek onda krene sa izvršavanjem. Problem može da se manifestuje na jako naprijatan način – definisali ste na primer *alias* za komandu DEL tako da brše fajl bez obzira na atribute recimo

```
alias dal = *del /z
```

Ako sada probate,

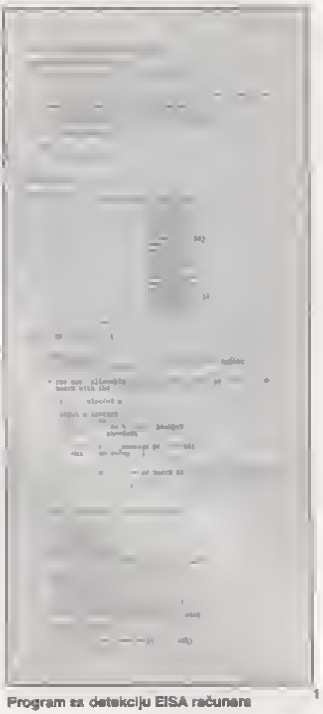
```
except (* zip) del * *
```

biće izbrisane i ZIP datoteke, tj. uradićete DEL * *, što ste svakako želeli da izbegnete [preuzeto sa Sezama, prilog **Nenad Batočanin** (*nbatocanin*)]

758 [ALGO] Službe za platni promet je kao što je poznato odlučile da od 1. oktobra svaki žiro-račun „opteretí" kontrolnim brojem pomoću koga bi trebalo da se eliminišu greške pri unosu brojeva računa (Službeni list SR Jugoslavije, 24/93 dopune u Službenom listu 57/93). Dokumentacija koja je tom prilikom ponuđena je, najblaže rečeno vrlo kriptična, ali algoritam u osnovi nije preterano složen i dosta liči na onaj kojim se generišu (prošle godine uvedene) kontrolne cifre brojeva faktura po modelu 05

Žiro račun najpre treba proširiti ne formu od striktno 15 cifara, i to tako da sa broj partije (deseti broj) popuni nulama sa leve strane. Recimo žiro račun 85500-601-2986 postaje 855006010000002986. Sada sa cifre, sa desne ulevo, množe sa 2, 3, 4, ... 16 i 15 i proizvodi sabiraju, dakle 6*2 + 5*3 + 9*4 + .. + 6*15. U datom primeru dobije se 467

Ostaje de se nađe ostatak pri deljenju zbira sa



Program za detekciju EISA računara 1

1, u datom slučaju 5. Taj ostatak se oduzima od 11 i dobija se konačni kontrolni broj 6. Što znači de bi žiro račun sa kontrolnim brojem glasio 85500-601-6-2986. Zanimljivo je da je kontrolni broj uvek jednocifreni – ako bi oduzimanjem 11-ostatak dobije 10 ili 11, kontrolni broj je nula [preuzato se Sezama, prilog **Dragan Grbić** (*dr grba*)]

Najveći zabune nastaje oko žiro-računa građana i raznih privatnih firmi kod kojih se također uvodi kontrolni broj: all se on odnosi samo na prve tri grupe cifara. Uzmimo za primer Beobanku – svi žiro-računi njenih štediše počinju računom Beobanke, koji je 60811-620-16. U skladu sa opisanim previlima, kontrolni broj je 1, pa žiro račun postaje 60811-620-1 16. Nastavak koji obuhvate vrstu računa ozneku filijale i sam broj partije građan ne sa naprosto **nedovezuje** na žiro račun banke sa kontrolnim brojem dakle mogući broj računa je 60811 620-1 16-51-353-5555555. Primetimo da čitav segment 51-353-5555555 name kontrolni broj, ako ga sama banke nije nekako predvidele kao dodatnu grupu cifara

Što se same korišćana funkcije tiče, ni ona nije naročita, tj. nije optimizovana tako da se umanji verovatnoća da sa dve greške pri unosu „poru" rezultirajući korektnim kontrolnim brojem ne nakorektno unesenom žiro računu

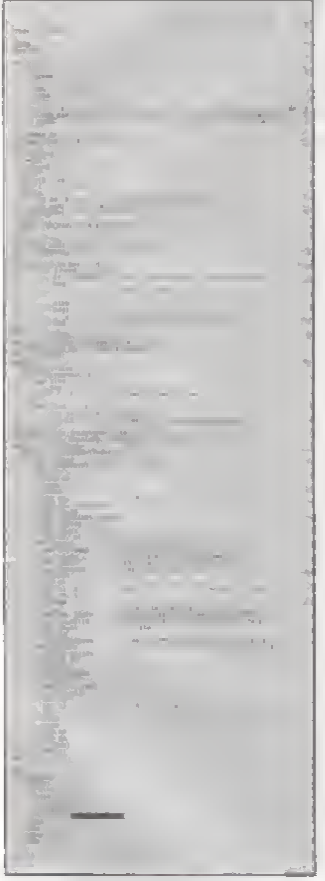
759 [BASE.CLIPPER; PC] Stavljale li „po automatizmu" SET DELETED ON na početak svih svojih programa? Ako vam nije preko potrebno filtriranje obrisanih slogova, preporučujemo da komandu prepravite u SET DELETED OFF običan *dbSkip()* po celoj bazi će ovako raditi 6-7 puta brže. Izgleda da *Clipper* SET DELETED ON prevodi kao SET FILTER TO Deleted() za svaku bazu što naravno usporava rad [preuzeto se Sezama, prilog **Jovan Bulajić** (*bulaja*)]

760 [BASE.CLIPPER, PC] Nedostatak debagera koji se isporučuje uz *Clipper 5.2* ako imate PRG fajlove u više direktorijume i poželite da otvorite neku datoteku koja nije u tekućem, treba samo da unesete njeno ime. Tako je bar, bilo u prethodnoj verziji nerazvno pod uslovom da ste postavili odgovarajući *Path for Files*. Novi, bitno „šareniji" debager kao de je izgubio ovo lepo svojstvo – more da se unosi puno ime datoteke sa sve „stazom" i varovali ili ne ekstenzijom ne podrazumeva se PRG [preuzeto sa Sezama, prilog **Nenad Batočanin** (*nbatocanin*)]!

761 [ZAST; PC]. Podaci o deset najčešćih (i najgorih) virusa.

- *Michelangelo* bnše prvih 9 megabajta inficiranog diska na Mikelanđelov rođendan 6. marte svake godine. Podaci se nepovratno gube. Ovim virusom možete de se zarazite podižući sistem se inficirana diskete
- *Stoned* ispisuje poruku „Your PC is Now Stoned!" svakog osmog puta kada resetujete (ili uključite) računar. Pod nekim okolnostima virus može de uništi sadržaj FAT-a, posle čega je jako teško (a često i nemoguće) oporaviti podatke ne disku. Prenosi se, kao i *Michelangelo*, podizanjem sistema sa zaražene diskete.
- *No-Int* je varijante *Stoned* virusa (zovu je i *Stoned 3*) koju je izuzetno teško detektovati u memoriji. Često se „svađa" sa nekim od uslužnih programa, e u tom slučaju obično strada *boot* sektor tabele particija.
- *Jerusalem* virus srećom nije najbolje napisan, pa se njegovo postojanje lakše detektuje. Svakog petka počinje da svima pesmicu *„Frere Jeques"* ne svakih 5 minuta, usporava rad komjutera, ieortave omi pravougaonik pole sate nakon što zarazi neki fajl, više puta se „kači" ne kraj EXE fajlova sve dok oni ne postanu *„too big to fit in memory"*. Tapi se za COM EXE, OVR OVL i slične fajlove, a prenosi se preko disketa, mrežnih diskova, senjskih veza, sistem zarazite startujući zaraženi program. Postoji više varijanti

Slika 2



Programiranje palete na VGA kartI

ovoga virusa koje se „razmnožavaju" na sličan način all im je krajnji cilj različit.

- *Azusa* se učitava kao rezidentni program i preuzima interapte preko kojih se komunicira sa diskom. Nakon što zaraza 32 diskete ekrembluje komunikaciju se portovima i pravi drugu štetu. U stanju je da totalno poremeti sadržaj diskete koju koristite. Pronosi se na kompjuter butovanjem sa zaražene diskete.
- *Joshi* blokira kompjuter svakog 5. januara, neprestano ispisujući poruku *„Type Happy Birthday Joshi!"* Prilikom preparavanja 5.25 inčnih disketa može učiniti kopiju neupotrebljivom. Prenosi se kao i *Azusa*.
- *Form* simulira klikTanje tastature kada je korisnik 18 i 24 dana u mesecu. Uništava fajlove na disku, kriptografišući ne neki način njihov sadržaj. Prenosi se kao i *Azusa*.
- *Dark Avenger* se instalira u memoriju i zaražava EXE, COM i razne *overlay* fajlove. Nakon svake šesnaeste uspešno obavljene infekcije briše slučajni sektor na disku, pa gde pogodi! Takođe i zaziva „ukrštene klastere" oštećuje FAT-ove, snižava performanse mreže. Zaraza se prenosi startovanjem inficiranog programa.
- *1575* virus ispisuje stub koji „jede" karaktere. Inficira COM i EXE datoteke ali na oštećuje njihov originalni sadržaj. Zaraza se prenosi startovanjem inficiranog programa.
- *Cascade 170X* deluje isključivo na COM fajlove – kada se neki zaražen program startuje instalira se kao rezidentni program. Povremano zablokira ili resetuje računar. U jednoj varijanti ovog virusa u slučajnom trenutku formatira disk [prilog sa *McAfee Associates* BBS-a, preneo **Nenad Nikolić** (*nikkus*)]

762 [TEHP PC; PC]: Da li ste nekada pokušali da programirate paletu VGA karte? Program sa slike 2 je dobar način da „zaronite" u ovu oblast – videćete kako sve to izgleda u tekst režimu [preuzeto sa Sezama, prilozi **Predrag Drljača** (*drpr*) i **Vladimir Klisić** (*klisa*)]

763 [COMM, PC]: Svaki iole noviji PC ima časovnik realnog vremena, obično zasnovan na Motorolinom čipu 146818 ili Intelovom 8253/8254 pa ipak svaki računar ne zna baš uvek koliko je sati – interni časovnici često žure ili kasne, a i nisu dovoljno precizni za mnoge specifične primene. Veća preduzeća obično zahtjeva nabavku neke posebne kartice a što se tiče doterivanja sata – može da pomogne i običan moderni

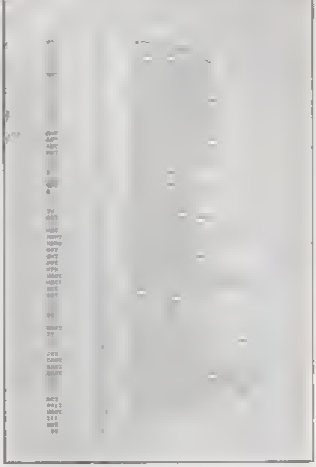
Počevši od 1966. godine američki Nacionalni institut za standarde i tehnologiju (NIST *National Institute of Standards and Technology*) održava poseban BBS u Boulderu, Kolorado. BBS je nazvan ACTS, što je skraćenica od *Automated Computer Time Service* – svako može da pozove ACTS i dobije izuzetno pouzdan i precizan (*do* granice milisekunde) podatak o vremenu, koji se održava na osnovu sinhronizovanog rada većeg broja atomskih časovnika i stalno proveravan astronomskim posmatranjima. Treba podesiti komunikacioni program na 300 ili 1200 bps, parametri veza 8N1 i pozvati +1-303-494-4774. Dobićete kratak ASCII lapis – prvi znak je zvezdica (marker), a zatim sledi vreme koje bi trebalo da se odnosi na trenutak kada je čitav ispis **stigao** do vas. To vreme je precizno proračunato preme brzini komunikacije, što znači da uređaj šalje tekst nekih 45 milisekundi **pre** nego što ispisano vreme nastupi (8 ms za slanje stringe na 1200 bps, plus 7 ms da signal stigne do prosečno udaljenog korisnika u Sjedinjenim Državama, plus 30 ms pauze koju unosi prosečan modem). Zvanja preko satelita i, uopšte prekookeanski pozivi unose dodatno kašnjenje koje dostiže oko 300 ms. S obzirom da je to vreme jako teško proceniti i da ono zavisi od mesta sa koga se zove, uveden je jedan trik – vaš komunikacioni program treba nekako „nagovoriti" da **odmah** po prijemu pošalje string nazad na osnovu čega će softver proceniti odstupanje i automatski mu se prilagoditi pre slanja sledećeg stringa. Posle uspešnog prilagođavanja, vodeća zvezdica menja se u „tarabu" (#)

Sat možete da doterate i preko *Internet*-a, kontaktirajući se sistemom koji se zove *time_a.timefreq.bldrdoc.gov* (npr. sa *telnet 132.163.135.130*). Međutim, ovakvi pozivi nude znatno manju preciznost, jer se ne može proceniti zadrška koju unosi mreža, osim toga, te zadrške se, zavisno od opterećenja mreže, menja iz sekunda u sekund

Rešenje za one kojima često treba superprecizno vreme je nabavka radio časovnika koji prima vremenske signale koji se neprekidno emituju u čitavom svetu. Ovakav uređaj košta, zavisno od modela i mogućnosti, između 500 i 5000 dolara, a raspoložive su i kartice za PC slične namene. Mana je što se često ukazuje potreba da se „razapne" antena da bi se signal mogao pouzdano locirati [preuzeto se BIX-a]

764 [COMM 764, PC]. Sistem u Arizoni će vam prirodno dati lokalno američko vreme koje tek treba „usaglasiti" sa našim. Srećom, uvek se dodaje

Listing 3



Ofseti vremenskih zona u odnosu na Grinič

časo broj časova, a minuti i sekundi ostaju nepromenjeni. Na slici 3 dajemo ofsete svih vremenskih zona u odnosu na Grinič – tu su i skraćenice koje ćete često pronalaziti u zaglavljima raznih poruka koje putuju elektronskom poštom, a za njih će vas pitati i instalacione procedure raznih operativnih sistema, na primer *Unix*-a [preuzeto sa Sezama, prilog **Dragan Grbić** (*dr grba*)]

765 [IGRE, PC] Autor mnogih novijih igara pokušavaju da se zaštite lozinkama – odjednom vas kompjuter pita šta piše u N-tom redu M-te strane uputstva i, ako ne znate, neme dalje! Pojavio se, međutim, *shareware* program kojim se lozinke mogu zaobići. Autor, prirodno upozorava da program nikako ne treba koristiti da bi se piratovali programi već samo kao pomoć da se igrač ne trudi oko listanja uputstava. Podržane igre obuhvataju *688 Attack Sub, 8 Ball Deluxe, A.T.A.C., Alone in the Dark, Battle Hawks 1942, Carrier Command, Castles II, Chess Maniac 5 Billion and 1, Civilization, Command HQ, Conquest of the Longbow, Darklands, Darkseed, David Leadbetter Golf, Dune 2 by Virgin, Eye of the Beholder III, F117A Stealth Fighter, F15 Strike Eagle III, Fields of Glory, Flash Back, Formula One Grand Prix, Hardball III, Indiana Jones and the Fate of Atlantis, John Madden Football, King's Quest VI, Lightspeed, Mike Ditka Ultimate Football (Part 1 & 2), Nuclear War, Pirates Gold, Planet's Edge, Prince of Persia II, Railroad Tycoon, Rex Nebular and the CGB Ring World, Star Glider II, Strike Commander (Part 1 & 2), Stronghold by SSI, The Legacy by Microprose, The Legend by Kyrandia, The Summoning, Tristan, Veil of Darkness, Where in the USA's Past is Carmen S., Wing Commander (I i II), X Wing Fighter* [preuzeto sa Sezama, prilog **Zoran Milosavljević** (*zormi*)]

Priloge za ovu rubriku šaljite na adresu „Računari" (sa „Bajtove lične prirode"). Duševar vojvoda Mišića 17, Beograd ili preko Sezama u vidu lične poruke uredniku (*mail write dejan*).

CODE NAME: SEZAM

Radovni čitaoci „Računara" avakako znaju da ja prvih dana novembra Sazam prosiavio četvrti rođendan – avečanoj prosiavi u „Elington" klubu hotaia „Hyatt-Ragancy" priaustvovaio ja preko 150 korianika i poaiovnih saradnika Sistema. Tamo smo doata govorili o Sazamu, njegovoj prošlosti, pianovima, budućnosti... neke od tih razgovora podeiićemo i sa čitaocima našag časopisa koja, varujamo, itakako zanima šta aa tokom protakia godine dešavalo na Sazamu. A dešavalo sa mnogo toga!

Oejan Ristanovic

Nadavno se jedna korisnica Sezama ponudila da napravi horoskop sistema – kada se već horoskopi prave za države firme i organizacije, zašto na bi mogli da se naprave i sa BBS! Ili su prirodno pobrbni podaci o ličnom vremenu i mestu „rođenja" Što se mesta tiče nema dilema – zgrada BIGZ-a negde između Sajmišta i Pašje. Ineći sprat, redakcija časopisa „Računari"... zapravo, Sezam sa tokom ova četiri godine samo jednom „pomeno", i to baš ova godina ali svega se dvadesetak metara. Što se vremena rođenja tiče stvari nisu tako jednostavne – korisnici nam svaka godine čestitaju bar po tri rođendana Sezam je naime, pušten u pogon 25 oktobra 1989 godine, eli su prvi dani njegovog rada bili rezervisani za konfigurisanje i „fino podešavanje" sistema – registracija novih korisnika nije bila omogućena. Samo tri dana kasnije 28. oktobra, pozvali smo *SysOp*-e drugih BBS-ova da posete, razgledaju hvala i kritikuju Sezam Vest se brzo pročula i korisnici su počeli da pristižu – pre nego što smo i shvatili šta se dešava, Sezam je imao stotinak korisnika a njegova telefonska linija bila je zauzeta gotovo tokom čitavog radnog vremena. Probni rad trajao je desetak dana a onda smo ga proglasili završenim sa potrebe izrada eventualnih horoskopa Sezam je zvanično počeo da radi u ponoć između 10 i 11 novembra 1989 godine

Razvoj Sazama bio je gotovo meteorski – već do kraja 1989 godine u konferencije je pristigio preko 1000 poruka a poziva je bilo toliko da smo vrlo brzo morali da predemo na 24-časovni rad i počnemo da razmišljamo o proširenjima Sezama i prelasku na komercijalni rad Taj važan događaj gotovo da se poklapa sa prvim Sezamovim rođendanom – oktobra 1990 godine Sezam je počeo da radi na tri linije i to na principu preplate. Mesec dana kasnije postali smo zvanični informacioni servis Šahovske Olimpijade u Novom Sadu i za tu priliku radili na 15 javnih linija Posle je Olimpijada završena Sezam je radio na šest linija sve do oktobra 1991 godine kada je broj korisnika toliko porastao da su četiri nove linije postale neophodne. Na 10 linija smo radili sve do prvih dana ove godine – na rasli broj korisnika zahtevao je novo proširenje kapaciteta Sezama na 15 javnih telefonskih linija! Ovim proširenjem Sezam je „uhvatio priključak" se najvećim evropskim BBS-ovima – najveći sličan sistem u Evropi, švajcarski „HITLINE Communications Ueli Maurer" ima 21 liniju

NOVI HARDVER ZA NOVE LINIJE

Iako su same telefonske linije glavni problem koji treba rešiti da bi se Sezam na ovaj način proširio (u pošti ih konstantno nema), izbor i nabavka opreme uvek predstavljaju najozbiljniju temu se razmišljanje Od samog početka Sezam je koncipiran kao lokalna *Ethernet* mreža PC računara. Ta mreža trenutno radi pod *Novell*-om 3 11 i sma dva servera – jedan od njih je *Compaq DeskPro 386* se diskom od 330 megabajta a drugi „klasičan" 386/33/8 M RAM se diskom od jednog gigabajta. Nema sumnje da dva servera obezbeđuju brzi rad mreže pošto se važni podaci mogu rasporediti na dva diska ali moramo priznati da to ubrzanje nije bilo glavni razlog koji nas je opredelio da u mrežu uključimo drugi server Više puta se naime, pokazalo da se *Compaq*-ovi računari mogu proširivati samo *Compaq*-ovom opremom koju u vreme sankcija nije lako nabaviti a ni platiti Zato smo *Compaq DeskPro* zadržali kao visoko pouzdani server koji obezbeđuje osnovne podatke (Sezamov softver, korisničku bazu tekuće konferencije i podsistema) dok se glavnina konferencija i programi u javnom vlasništvu smeštaju na drugi server se četvorostruko većim diskom!

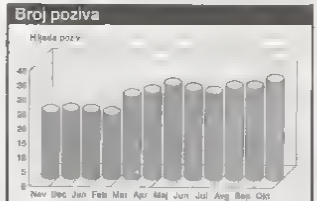
Stanica su 15 AT i 386SX računara opremljenih *Etherlink* kompatibilnim mrežnim karticama i modemima *ViVa 2400 MNP* i *Discovery 2400 MNP* Što se *ViVa* modema tiče pokazali su se izuzetno kvalitetnima i pouzdanima ali ih opet zbog sankcije nije lako nabaviti. Zato smo početkom 1993 godine sistemu dodali jedanaestu liniju i na njoj testirali nekoliko tipova modema, procenjujući pre svega njihovu pouzdanost i kompatibilnost sa modemima korisnika. Što se pouzdanosti tiče, iskustvo nas je naučilo da se većina modema u uslovima neprekidnog reda povremeno „zaglupljuje" – modem dođzi u stanja iz koga se može povratiti jedino isključivanjem napajanja, što na pravi posebne smetnja ako je korisnik pored njega, ali je sasvim neprihvatljivo za sistem koji treba da se održava uz što manje ljudskih intervencija Što se kompatibilnosti tiče pokazuje se da modemi nekih drugih – kolieje su retke i malo bitne za individualne korisnike ali ako na neki BBS stavite modem koji je nekompatibilan se bilo kojim poznatim modsmom onemogućavate svim visnicima tog modema da koriste usluge BBS-al Posle dataljnog testiranja pokazalo se da *Discovery 2400 MNP* modemi zadovoljavaju najstrožije uslove pa smo se opredelili za njih I nismo pogrešili – se modemima nije bilo nikakvih problema. Čak se posleonjih dana pokazuje da *Discovery* bolje komunicira se bržim (9600 i 14400 bps) modemima od *ViVa*

Izbor stanica je bio nešto lakši – opredelili smo se za Jugodatine AT i 386SX računara koji su umesto disk jedinica opremljeni *boot ROM*-ovima. U početku je izgledalo da će se *boot ROM*-ovi teško složiti sa Sezamom – ako se recimo zbog nestanka struje čitava mreža isključi stanice se *boot ROM*-om će se prve pobrinuti i pokušati da uspostave vezu se serverom Ako server (koji treba da testira disk, startuje *Novell* i uradi mnoga druga operacije) nije spreman, stanica ispisuje poruku da server nije nađen i staje Da je spred stanice čovek on bi se svakako setio da sačeka neki minut i pritiena Ctrl Alt Del pa bi sve bilo u redu pošto čoveka nema, nod bi nsteo naupotrebljiv sve dok neko ne intervenše. Rešenje ja na sreću nađeno u malo modifikaciji *boot ROM*-a (kada će se računarstvo konačno oslobodili potrebe za ovakvim „hakenesanjem"?) – seda se računar automatski resetuje sve dok na „pronađe" server Na taj način smo uštedeli disk jedinica kontrolere tastature i monitore, a postoji i ideja da se više Sezamovih stanica montira u istu kutiju čime bi se uštedela i posebna kućišta odnosno napajanja, za seda na kome nismo radili ali kada bude trebalo da se dodaju nove linije

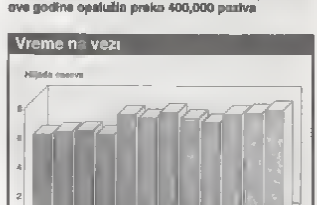
Instalacije novog hardvera bila je dobra prilika da malo „protresemo" Sezam proverimo spremu zamenimo sve mrežne kablove koji su nam povremeno donosili probleme obnšemo i ušlešmo prašinu i uopšte da učinimo ave što se može učiniti da Sezam ubuduća radi što pouzdanije Tom prilikom Sezam smo preselili u drugu prostoriju u kojoj je zahvaljujući pogodno izvedenoj instalaciji njegovo održavanje znatno olakšano „Operacije seoba" trajala je nekih 6 časova (što je najzanimljivije Sezam je tokom gotovo čitavog ovog intervala radio. doduše se zmatno smanjenim kapacitetima) Rezultat je današnji Sezam na 15 javnih linija

Termin **15 javnih** linija znači da na Sezamu povremeno možete sresti i više od 15 korisnika popodne i noću aktivna je posebna linija za održavanje sistema, a u mrežu je vezano i nekoliko redakcijskih PC računara se kojih tekoda može da se pristupa Sezamu, dobar deo komunikacije potrebna da se pripremi svaki od brojeva „Računare" obavlja se elektronskom poštom! Najzad jedan od Sezamovih servera povezan je sa lokalnom (*Arcnet*) mrežom Novinskog sektora BIGZ-a, pa je sistem postupačan i zainteresovanim novinarima drugih redakcija. Sve u svemu lista koja se dobija komandom WHO često se sastoji iz 18 ili 16 imena a „apsolutni rekord" su bile 22 korisnika na Sezamu iapis koji daje vrlo uskoro će morati da se podeli u dva kolona.

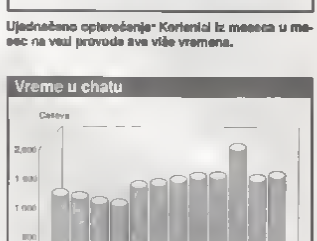
Osim pet novih linija, 1993 godina donela nam je i novi strimer koji obezbeduje znatno veću sigurnost podataka. *Backup* na traka se obavlja periodično (na bilo kakvo da ga pravimo i češće) dok se neki važni podaci (naročito korisnička baza) dodatno kopiraju na diskete ti na neki od lokalnih diskova u Redakciji. Ostale zamene hardvera sveia su se na tekuće održavanje – jedna mrežna kartica se pokvarila jedan nod se poneašao jako čudno pa je morao da buda zamenjen dodali smo nešto memonje u jedan od servera isprobavali razne brža modeme na sysop nodu i tako daije Obzirom



Svakega minuta pa CONNECT Sezam je tokom ove godine opsluživ preko 400,000 poziva



Ujednačeno opterećenje: Korisnici iz meseca u mesec na vezi provode ava više vremena.



Avgust kao vreme sa mala nočne razgovora: vreme koja korisnici provode u chat-u.

na to da sve oprema radi 24 časa dnevno 365 dana u godini vrlo smo zadovoljni njenim funkcionisanjem kao i količinom „sportske sreće" koja prati naše napore da sve besprekorno funkcioniše

Povremeni problemi se hardverom uglavnom su posledica stvari nad kojima nemamo kontrolu – par puta je nestajalo struja (server su naravno podržani UPS-om, uređajem za nepreskidno napajanje ali sve stanice trenutno „popadaju"; ponekad se desi da je za ponovno aktiviranje neka od njih potrebna ljudska intervencija) sve slabije grejanje zgrade u zimskim mesecima negativno se odražava na rad opreme pojedine linije su greškom pošte „lipadale" iz senja ponekad problem smo i sami „skrivili" eksperimentišući sa redekcijskom granom naše mreže Srećom svi problemi ovoga tipa otklanjani su jako brzo tako de se može reći da Sezam tokom čitave godina bio van pogona oko 24 sata računajući tu i razne najavljene prekide rada zbog održavanja sistema i instalacije nove opreme Takvi rezultati mogu da se ocene samo kao veoma visok stepen pouzdanosti ali se ipak radi na daljim unapređenjima Naročito su interesantne ideje o specijalnom hardveru koji bi obezbedio automatsko „leviačenje" sistema iz neke neregularne situacije har diverskim resetovanjem pojedinih čvorova Sledeći stepen bi bila poluautomatska operacija koje ovlašćena lica može obavljati posredstvom modema, verovatno do dana kada budu smišljani dovoljno inteligentni robot načemo modi da izbegnamo potrebu za „ručnim" intervencijama u nekim prilikama, ali se bar trudimo da ih svedemo na što manju meru.

SOFTVERSKA KONCSPCIJA

Sezam je kao što je poznato zasnovan na domaćem softveru – glavni program napisao je Zoran Životić Dobro promišljena softverske koncepcija svakako je mnogo doprnela uspehu Sezama u centru pažnje uvek je bila potreba da konsnik sveku od željanih operacija obavi što brže i efikasnije a da se dobar deo ih operacija „prabaci" na korsnikov PC tj da se sa Sezama samo pokupe nove poruke i pošalju odgovori a da se njihovo čitanje i pisanje obavlja *off line* bez opterećenja telefonskim impulsima koji stalno „kucaju" Sezam se oslanja na transfer protokole (uglavnom *ZModem*) kako bi se umanjili problemi koje donose loše telefonske veze – poruke i odgovori prenosa se u arhiviranom obliku a transfer protokol po potrebi ponavlja pojedine segmente dok se na ustanovi da je datoteka verno prenesena. Uz sve to korsnički interfejs prilagođen je kako apsolutnim početnicima tako i veoma iskusnim korsnicima modema – prvima je dovoljno da izaberu *Automatski režim* i docnije pregledaju datoteku koja im je stigla a drugi su u mogućnosti da veoma precizno odrade šta će praititi i šta neće da pretražuju direktorijuma i konferencije u potrazi za željenim programima..definišu svoje sopstvene komande i skraćenice i uopšte prilagode Sezam kako svojim potrebama tako i svom ukusu

Sezamov softver naravno nikada nismo smatrali završenom pričom – program koji danas kontroliše sistem veoma je različit od onoga se kojim smo počeli i iako smo naravno sve vreme sledili istu koncepciju i obezbedili visok stepen verbilne kompatibilnosti Veći deo izmena rezultat je sugestija samih korsnika – konferencija SEZAM uvek je puna poruka u kojima se predlažu razne unapređenja. Treba ipak reći da tokom ova godine nije bilo značajnih izmena u Sezamovom softveru – glavni program ostao je isti a instalirano je nekoliko eksternih programa pomoću kojih smo „obezvili" par nagradnih igara i obogatili Sezamovu ponudu novim uslugama kao što je kurirska tiska Zapravo Sezam je nekako tako i zamišljen – glavni program treba da obezbedi komunikaciju sa korisnicima i osnovne usluge konferencijskog sistema a sve ostalo bi bilo prepuštano eksternim programima. Dalji razvoj te koncepcije podrazumeva usavršavanje osnovnog programa koje smo započeli još sredinom prošle godine – treba obezbediti da on bude što modularniji i lakši sa održavanje uz očuvanje visoke pouzdanosti i po mogućnosti unapređenje performansi. Ostvarenje ovih prilično ambicioznih planova egleda zahteva više programerskog rada nego što smo mislili – revizije softvera, planirana još za kraj 1992 godine, još nije završene Posao je produžila odluka da se promeni kompletan sistem rada sa datotekama na nekom nivou tj da se celokupne na *BTrieve* – procenili smo da ćemo na taj način povećati pouzdanost olakšati održavanje sistema i značajno unapredit performanse. No ovako korenite izmene zahtevaju i dosta testiranja, što opet traži vreme tako da se za sada ne može predvideti kada će nove verzija softvera biti pokrenuta. Ostaje nam da se nadamo da će je korsnici oceniti bar onoliko povoljno koliko i sve prethodne

Tokom ove godine na domaćem tržištu se pojavile i par BBS programa koji što je veoma indikativno u prilično men kloniraju Sezamov korsnički interfejs i komandni jezik. Jedan od takvih programa napisao je Dragan Zakić i pod njim je radio *Profile BBS*. Drugi je napisao Stanislav Šokorac za potrebe *Pingvin BBS* a Činjenica da se domaći autori odlučuju za kloniranje Sezama svakako je veliko priznanje koncepciji koju smo usvojili ali i rezultat renomea Sezama među vlasnicima modema u Jugoslaviji – zahvaljujuća dobroj dokumentaciji i podnici toliko ljudi je naučilo naš komandni jezik da to ni jedan autor BBS softvera na može ignorisati

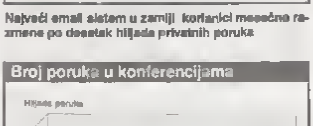
SEZAM U „GLOBALNOM SELU"

Tokom 1993 godine Sezam je napravio veliki korak prema „globalnom selu" kako mnogi nazivaju svet računarskih komunikacije Opšti trend je umrežavanja i to na rasnim nivoima – u najmanju ruku treba omogućiti korisnicima jednog sistema da šalju elektronsku poštu korsnicima neke drugog a kasnije se očekuje i prenos datoteka direktno prijavljivanja na druge sistema I tome slično Tokom prvih Sezamovih godine uglavnom smo se trudili da razvijeme naš sistem, ova godina došlo je na red i povezivanje se drugima!

Broj privatnih poruka

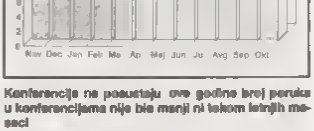
Najveći email sistem u zamlji korisnici mesečne razmene po desetak hiljada privatnih poruka

Broj poruka u konferencijama



Konferencije na posustaju ove godine broj poruka u konferencijama nije bio manji ni tokom letnjih meseci

Download datoteka



Sve veće interesovanje korisnici sve više download-uju javni softver sa Sezama

impuls u ovom smeru dalo nam je formiranje fleointernela mreže u koju je uključeno više računara na Fakultetu Organizacionih Nauka (FON) i Elektrotehničkom fakultetu (ETF) u Beogradu Ubrzo smo ostvarili vezu bazirano na UUCP protokolu koje obezbeduje razmenu elektronske pošta između korisnika Sezama i korisnika bilo kog sistema vezanog u Beointernet Omogućili smo i razmenu pojedinih konferencija, tako da se trenutno eksperimentalno razmenjuje Sezamova konferencija MALI OGLASI se konferencijom *yu.oglasi* Beointernela uskoro će se razmenjivati još neka stručne konferencije

Glavne prepreke punoj vezi se svetom su sankcije zbog kojih je Jugoslavije odsečena od *Bitnet*-a i zbog koja nemamo registrovani *Internet* doman Jedine modemska veza se svetom kle poštanskim (JUPAK) linijama, i baš se tim putem, primenom UUCP protokola razmenjuje elektronska pošta se svetom Na žalost, mala propusna moć ovog kanala ne omogućava da on bude otvoran za sve korisnike Beointernela.

Razmišlja se i o drugim vidovima povezivanje Sezama se kompjuterskim mrežama u zemlji i svetu – pre svega o JUPAK priključku, a „prava stvar" bi bila linija koja nas direktno povezuje sa nekim sistemom veza nim u globalni *Internet* tek bi tada naši korisnici na način opisan u prošlim „Računarima" dobili mogućnost da se komandom *telnet* direktno povežu se nekim računarom u svetu ili da recimo komandom *finger* pogledaju koliko ima konzervi Coca Cole u automatu nekog američkog univerziteta Verovatno ćemo za pun pristup globalnom *Internet*-u morati da sačekamo ukidanje sankcija, ali u međuvremenu može dosta toga da se uradi Važno je da smo napravili prvi korak i da korsnici već uživaju u blagodetima UUCP pošte.

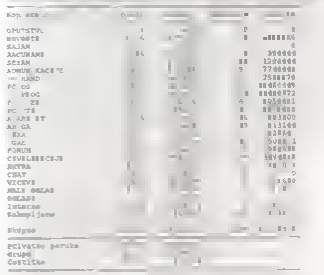
GODIŠNJI IZVEŠTAJ

Cifre u Sezamovom godišnjem izveštaju zbilja su impresivne – od 11 novembra 1992 do 11 novebra 1993 godine opelušano je 403762 poziva u prosteku se na svakih minut i dvadeset sekundi uspešno uspostavila po jedne veze! Ukupno vreme komunikacije sa korisnicima bilo je 86178 5 časova što znači da je nepreklidno bilo zauzeto gotovo 10 nodova! Za tih 86.000 časova (3590 dana) korisnici su u konferencije poslali preko 111 000 poruka ili 114 megabajta teksta, 77.500 privatnih poruka (39 megabajta) *upload*-ovali 800 megabajta i *download*-ovali čak 20 gigabajte podataka! Ovakvim skorom može da se pohvali reiko koji evropski sistem

Korsnici su nas najviše zvali u oktobru najviše vremena na vezi proval u maju. za konferencijska rasprave bili najzainteresovanji u septembru najviše privatne pošte stglo je u avgustu kada se najviše i *chat*-ovalo Posebno je zanimljiva krivulja koja opisuje *download* Sezam je zamišljen kao konferencijski sistem tako da smo u prvim danima posvećivali manje pažnje direktorijumima trudili smo se de probleremo najbolji *public domain* softver iz nekoliko oblasti i ponudimo ga korisnicima No to su bila druga vremana – veza se *Bitnet*-om je perfektno funkcionisale pa su se se *Trickle*-a iako naručivali svi postojeći javni programi Od leta 1992 godine nabavka javnog softvera počela je da predstavlja problem što je zahtevalo pojačani angažman Sezama na tom polju U saradnji sa kolegama se Imtal BBS-a i FON File Servera, organizovali smo nabavku najnovijih javnih programe tako da su naši korsnici dobijali na samo SCAN CLEAN PKZIP i DSZ, nego i mnoga druge interesantne programe svega nekoliko sati posle svetske promocije Pošto se stručna literatura sve teže nabavlja, pažnju smo posvetili i nabavci svih materijala do kojih može da se dođe u elektronskl čitljivom obliku – naši korsnici mogu da prezumu opise raznih formata precizne definicije svih (dokumentovanih i nedokumentovanih) PC-jevih INT ova *Norton Guide* datoteke za mnoga popularne programske pakete nešto prostora i rezervisano je i za nekompjutersku literaturu pa i za beletristiku koja je na *Internet*-u svakim danom sve više Jedina bitna postavljena je pred komercijalni softver Sezam je od prvog dana vrlo rančna da se u direktorijumima I konferencijama na pojavljuju bilo kakvi sadržaji koji bi ugrožavali autorska prava domaćih i stranih softverskih firmi.

Korsnici su veoma zainteresovani se nove arhive u direktorijumima što svakako pokazuje i grafik na kojima je prikazan mesečni *download* od uvođenja novih linija umalo se na nivou između 1.5 i 2 gigabajta mesečne, uz „pikove" u maju i oktobru

Sezam u prošlој godini



Gotovo sve o Sezamu

Gotovo sve o Sezamu: statistika Sezama tokom protekle godine

SEZAMOVE KONFERENCIJE

Nova linija i veći broj poziva uvek donose nov kvalitet konferencijama što se potvrdilo i ova godina – poruka na samo da je bilo mnogo vila već je i kvalitet diskusije posebno u stručnim konferencijama bio na solidnom nivou. To naravno ne znači da nije bilo oštrih polemika pa i nepotrebnih sučaljavanja, ali smo već navikli da povremeni „incidenti" prate žešće rasprave ogroman deo poruka je ipak, ostajao u okviru demokratskih okvirima. Ugašenih konferencija nije bilo a pokrenuli smo i mnogo novih tema. No podimo redom

NOVOSTI

Konferencija NOVOSTI „odgovorna" je za gotovo 10% Sezamovih konferencijskih poruka to je zaprava segment namenjen informisanju korisnika Sezama o onome što se dešava u svetu. Najviše informacija se, prirodno odnosi na računarsku scenu – hardver softver usluga događaji trendovi .. vesti gotovo svakodnevno stižu sa BIX a, *CompuServe* a i drugih sistema. Korisnici su medjum pokazali interesovanje i za nekompjuterska vesti informacije iz domena politike kulture sporta i tome slično Aleksandar Bakić (*bale*) i Srđan Kusovac (*squsovac*) uložili su mnogo napora da pronađu strane sisteme se kojih se mogu preuzimati ažurne vesti te likera najpoznatijih stranih agencija – ve ali svakodnevno stižu na Sezam I „slažu" se u temu *world.news*

Činjenica da je **dnevno** stiglo u proseku 183 kilobajta NOVOSTI praćenje svega bilo je interesantno tek za ponekog korisnika. Zato je sredinom godine konferencija NOVOSTI dobila temu *digest* u koju šaljemo po prvih par redova svake vesti zajedno sa brojem poruke koja nosi kompletan tekst. Stanislav Šokorac (*ssokorac*) je napisao program koji olakšava praćenje *digest*-a i izbor poruka koje želite preneti na svoj računar

SEZAM

Konferencija SEZAM bavi se našim sistemom – to je mesto na kome početnicima pomažemo da se snađu na koma iskusniji korisnici predlažu izmene Sezamovog softvera, tu obaveštavamo korisnike o novitetima na sistemu tu oni ukazu probleme na koje su naišli ukratko u konferenciji SEZAM nikada nije nedostajalo poruka!

Događaj godine u ovoj konferenciji svakako je proširenje Sezama na 15 linija, veliki broj poruka izazvala je i emisija „Codename Sezam" emitovana na NTV Studio B povodom Sezamovog rođendana kao i novitet u softveru – bazu kurseva stranih valuta zanimljivi citati kao i nagradna igra koje smo organizovali pra Nove godine i Dana zaljubljenih

Što se tema *problemi* tiče najviše poruka je pristiglo u julu povodom pojave koju su korisnici nazvali „mistnija noda 15" (više o njoj u „Računarima 94" Do-

ste poruka stiglo je i u temu *vijge* u okviru koje diskutujemo o vilekomunikaciji igrama na Sezamu najviše se razgovaralo o mogućnosti instalacije programa *Trade Wars* a propraćeno je i tridesetak partija domaće vilekorisničke igre HANGMAN Najveće interesovanje izazvala je partija u kojoj je cilj bio pogađanje recenica koje bi korisnici ugovorili kada bi navino šetajući beogradskim ulicama, naletali na *Kim Basinger*

KOMUNIKACIJE

Konferencija KOMUNIKACIJE bavi se računarskim (a pomalo i ostalim) komunikacijama modemima popularnim komunikacionim programima, protokolima, elektronskom poštom domaćim i stranim BBS-ovima i informacionim sistemima i sličnim temama. Prateći poruke u ovoj konferenciji stekli smo utisak da su domaći korisnici računara tek početkom 1993 godine počeli da „otkrivaju" i proučavaju teks modema – poruka je sve u svemu bilo tolika da su teks modemi tokom godine dobili posebnu temu

Kada se radi o komunikacionim programima najviše interesovanje izazvao je *Procomm Plus for Windows* No mnogi su i dalje vemi komunikacijama pod DOS-om a tu je najpopularniji *Telemate* Tokom 1992. godine sprobali smo *Telemate 4.0* program koji je dobio mnogo komplimenata ali i namučio stotine korisnika, pra svega njihovom sopstvenom krivcom zbog nelegalne „registracije" I poruka tipa *Prase $ if you understand shareware concept*

Tema *email i ljupak* uglavnom su se bavila mogućnosti slanja elektronske pošte pod sankcijama, najčešće poštanskim kanalima Tema vax bavila se VMS-om i njegovom specifičnošću komunikacijama između domaćih VAX-ova i drugih računara elektronskom poštom na VAX-u I mnogim drugim stvarima Najviše poruka izazvali su vesti o „provalama" u rake od beogradskih VAX-ova – zaključak je bio da **nikako** na insba otvarati svoj osnovni katalog za *write*

Tema *strani sistemi* najviše se bavila *Internet-om*, *Usenet-om* i njihovom mogućnošima mnogi korisnici su naši načina da se preko JUPAK-a, „dokopaju" globalnog *Internet*-a ali im je hronično nedostajala literatura o njegovim mogućnostima i servisima Što se tema *yu bbs* tiče pokazalo se da nije lako ni prikupiti podatke o svim domaćim BBS-ovima, a kamoli steći uvid u usluge koje pružaju – tokom čitave godine smo se trudili da što češće ažuriramo spisak domaćih BBS-ova i drugih informacionih sistema a opet se on stalno pokazivao neažurnim jer su nova BBS-ovi nastajali i nestajali iz dana u dan Već sistemi koji su *takoreći* osnivali 1993 godinu pa sve preko FFS FON *File Server* i, nešto drugo je *Seointernet*

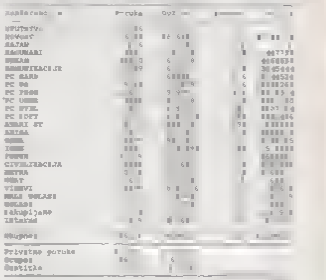
Na zaboravimo ni temu *hackers koja* je uz ostalo, donela nekoliko izuzetno zanimljivih knjiga – sve je počelo od čuvenog romena Kliforda Stola. (*Clifford Stoll*) *Cockoo s Egg* romana (bazirano na istinitim događajima) o nemačkim nekrima koji su proveljivali u računara Američke armije CIA-e NSA i drugih agencije skupljajući podatke koje su docnije prodavali KGB-u Otkriveni su kada je jedan fičar primetio grešku od 75 centi u softveru za obračun cena kompjuterskih usluga Koji je naplaćivao, počeo da traga za greškom i naišao na nešto čudno Interesovanje koje je ovaj roman izazvao nalerala nas je da potražimo slične tekstove tako da su se na Sezamu ubrzo pojavile razne opsežne diskusije i priče tog tipa, da bi na kraju kao prava poslastica, stigao i *Terminal Compromise* Winna Schwartau-a, prvi „pravi" roman iridat u *shareware* obliku koji se distribuira preko *Internet* a. Mnogo poruka izazvalo je i „domaće kukavičje jaje" ij hrada baze komanika se jednog domaćeg BBS-a koja je mnogima donela razne neprijatnosti Pokazalo se da tu i nije bilo nekog većeg hakerisanja – *SysOp* je bio nisopazan pa je gost prekopirao podatke na disketu

Što se nekompjuterskih komunikacija tiče u temi *ntvpt* diskutovali smo o preseku na tomako biranja ustanovišli preslano koje centrale u Beogradu (i Srbiji) to obezbeđuju a koje ne. Redovno smo pratili (i komentarisali) promene cena PTT usluga, a pred kraj godine saznali i za promene međunarodnih telefonskih tekska i JUPAK brojeva bivših jugoslovenskih republika – da bi se na primer Sezam zvao iz Amerike, sada treba „okrenuti" +381 11 848422.

PC.HARD

Konferencija PC HARD bavi se hardverom PC računara – mikroprocesornima matičnim pločama, diskovima interfejsima štampačima. Obzirom da je tokom ova godine bilo nekoliko veoma značajnih događaja

Sezam od kad postoji



Čitav „život" u jednoj tabeli: bilans Sezama od prvih dana rada do danas

na polju hardvera personalaca, i konferencija je bila bogata porukama Detaljno smo propratili prve informacije o *Pentium* u proučavali prve ploče se (dodušte praznim) godnoćlima za „586" poredili ISA, EISA i ostale magistrale proučavali keš kontrolera koji najbolje seradiju sa njima videli od kakvog je značaja brzine diska i isplati li se investicija u neki od najnovijih (i najbržih) modela radovali se jeftinijim diskovima i bmull zbog skoka cena RAM čipova.

Konferencije PC HARD se naravno nije bavila samo vrhovima PC tehnologije – većina poruka odnosi se na konkretne probleme na koje korisnici nailaze konfigurišući svoj 386 ili 486 računar U temi kvarovi pokušavali smo da pomognemo onima čiji se računar na ponaša kako se očekuje Neočekivano veliki broj poruka odnosio se na štampače naročito laserske. Očigledno je da sve veći broj korisnika ima potreba za izuzetnim izgledom dokumenata pa je najpra *HP LaserJet IV* a zatim i njegovim „izotopi" kao što su 4L, 4M i 4P izazvali prave lavine poruka Redovno smo nabavljali *Windows* drajvere za razne štampače ali problem konličenja domaćih slova (koje su ugrađiana u ROM *LaserJet*-a IV) iz *Windows* a i dalje ostaje otvoren

Konferencija PC HARD bavila se i širim dodlacima za PC– posebno je u sećanju ostala rasprava o džojstcima vođena početkom 1993 godine Pretresena su sve mogućnosti nacavke pa čak i samogradnje i rešeni razne problemi pri upotrebi nje ni čudo da su se podsetnim džojstcima, konemci Sezama bili pol puno spremni za razna takmičenja u letenju organizovana u okviru konferencije IGRE!

PC.OS

Polito se bavi operativnim sistemima se PC računara konferencija PC OS se tokom protekle godine svakako nije mogla „žaliti" na manjak tema za diskusiju lakšo je DOS 6 0 OS/2 2 1 *Windows NT* a nedavno i beta verzije *Chicage* budućeg *Windows*-a 4 0 Svaki od ovih događaja propraćen je detaljnim komentarima i mnoštvom polemika pri čemu je najviše interesovanja izazvao DOS 6 0 što je i prirodno kada se zna da većina korisnika, iako „jednim okom" osmatra novije operativne sistema u svakodnevnom radu i dalje koristi batenju DOS– *Windows* Najviše pažnje je izazvao DBLSPACE– nije bilo mnogo problema se njegovom pouzdanošću (premda je diskusija o tome smeju li se *doublespace*-ovani diskovi kešurati otkrila neke neobične stvari) performanse su pozitivno ocenjene ali su neki ipak ostali pri Stacker-u Slično je „prošao" i MEMMAKER koga QEMM i 386MAX prevazilaze kvalitetom

Što se tema *windows* tiče, „evergrin motiv" je svakako potraga za drajverima za razne specijalna štampače i SVGA kartice – vala nije cilj samo imati drajver nego postići optimalna performanse Aktuelan je bio i problem YU slova – predložena su razna rešenja za prepravku fontova a pokazalo se da je najbolje prepravku lebegavati jer se u svim editorima koji su nam

bili na raspolaganju gube neke od informacija (*hints*) o fontovima, mnogi su najzad odlučili da koriste „nelinux" fontove iz paketa *Windows for Eastern Europe*

Korisnici *Unix*-a bavili su se sličnostima i razlikama između raznih verzija ovog operativnog sistema koje su prilagođene PC računarima, mučili se sa instalacijom internih modema na „nereguliame" portove, a posebno interesovanje izazivali su problemi umrežavanja *Unix* (i drugih) sistema, i to kako na lokalnom tako i na globalnom nivou. Upoznali smo TCP/IP protokole i segledali mogućnost saradnje *Unix* računara se posebno čom *Novell* mrežom tj. vezama TCP/IP a i IPX-a. Zanimljivo je da *Novell 4.0* gotovo i nije pomenut u diskusijama – izgleda da još nema naših korisnika koji su odlučili da svoju mrežu prebace na ovaj operativni sistem.

Za bar 2000 od 5800 poruka koje su pristigle u konferenciju PC OS tokom proteklе godine „kriva" je velika septembarska diskusija o operativnim sistemima – sve je počelo od poređenja procesora i nagnavex o budućnosti RISC arhitektura da bi se preko problema vertikalna kompatibilnost prešlo na teren PC računara. Ogiasili su se i ljubitelji *Mac*-a, pa i Atarija. Šta je dobro a šta loše u DOS-u *Windows*-u, *Unix*-u *MacOS*-u, TOS-u da li je *Windows* operativni sistem ili obićan grafički *shell* da li korisniti GUI ili komandnu liniju da li *Windows* otkriva ono što na Meku već odavno otkriveno... Ovo su bile samo neka od mnoštva pitanja na koje su korsnici pokušali da daju odgovore. U diskusiji je bilo mnogo isključivosti i navijačkih poruka, ali kada su se strasti smirile i diskusija utihnula, ostao je i važan rezultat u vidu obilja interesantnih i korisnih informacija o raznim operativnim sistemima

PC.PROG

Konferencija PC.PROG namenjena je programerima kojih kako izgleda, predstavljaju veoma brojan segment Sezamove populacije. Tematika kojom se konferencija bavi vrlo je široko postavljena od dizajna softvera i algoritama preko programskih jezika koji su na PC-ju najpopularniji od programiranja niskog nivoa, asemblera, sistemskih poziva. Sredinom ove godine konferencija je dobila i temu *windows* koja se bavi isključivo pisanjem *Windows* programa.

Tema *algoritmi* nije se preterano bavila „klasikom" kao što su sortiranje i pretraživanje. Usredsredili smo se na „ovozemaljske" probleme koje su donele promene propisa – brojevi faktura, žiro računi i mnoge druge cifre koje se javljaju u raznim poslovnim aplikacijama dobile su tokom ove godina svoja kontrolne sume pomoću kojih treba da se umanji verovatnoća grešike pri unosu. Opisi generisanje kontrolnih brojeva koje je SPP (bivši SDK) dala kao da su deo Kremanskog proročanstva, pa je bilo desta muke dok je sve to protumačeno isprogramirano i isprobano na primerima

Porukama je standardno najbogatija tema *clipper* – na kaže se uzalud da 80% domaćih PC programera živi od ovog jezika. Ove godine je bilo puno raziloga se obilje poruka, pojavio se najpre *upgrade* za *Clipper 5.01* a onda i *Clipper 5.2*. I jedan i drugi proizvod donaljno su analizirani ali je prav „kvantni skok" donio *Blinker*, superbrz linker koji znatno ubrzava i olakšava razvoj *Clipper* programa i nudi brojne korisne funkcije i procedure

C programeri su podelili naspavu u dve teme – ccec se bavi „klasičnim" C-om a *cpp* C++-om. Pokazalo se da je interesovanje za prvu znatno prevazilazilo interesovanje za drugu – veliki broj poruka odnosio se na instalaciju i upotrebu popularnih C kompajlera, na zne probleme na koje se nailazi grešike pri optimizaciji, korišćenje popularnih biblioteka. Naročitu pažnju iza zvao je *Paradox Engine* koji nudi lak (?) način da se na C-u pišu eliksam programi koji rade se bazama podataka. Pravi bum poruka je međutim izazvan tekstom „The case against C" jednestenim se stranih mreža. Autor ovog, po opštoj oceni izvanredno napisanog teksta odporava mnoge kvalitete C-a, smatrajući razna njegova rešenja nespojivim se tehnjom da softver koji nasta je buda pouzdan i lak za održavanje kao primer „uzornog" jezika predlaže se *Modula-2*. Mnogi korsnici su počeli da osporavaju ili osnaže neke od iznetih argumenata, što je prerasio u diskusiju o programskim jezicima uopšte, a kasnije i o tehnologiji izrade softvera.

Temu *pascal* je obeležio ogroman broj *unit*-a u javnom vlasništvu koje je sa stranih mreža preneo Ivan Skerl (*iskerl*) – bilo je tu potprograma za svakoga, počevši od kontrole ekrana, zvuka, miša, komunikacionog interfejsa od kontrole modema komunikacija, prelaska diskiu na najnižem nivou... na kraju ano „morali" da o-

Najkorisnici

```
Najv Ko    s Ka

Kb  Ime i Ko                  Z ag ne    ne                    m                    Poslje

1   Ne  nio          Ov e  Ra  nda        BEOGRAD                                  10
2   Ba  j            Ko n  Ta  gra        BEOGRAD              BEOGRAD             101
3   bulaja           Jova  s  ar          BEOGRAD              BEOGRAD             801
4   ppaškaj s        Sla a  Ra  ić         NOVI BEOGRAD         NOVI BEOGRAD        1111
5   Ma  ina          Ranka Ra  ana  i      BEOGRAD                                  8117
6   M ova            S     A    ssssov     BEOGRAD                                  1111
7   Ma  ina          Ne ad  Ba  ić         BEOGRAD              BEOGRAD             1116
8   ncuilo           Vlada Cal  c          BEOGRAD              BEOGRAD             1001
9   Ivanka           Ljubom Pe  ss          NOVI BEOGRAD         NOVI BEOGRAD        1179
10  Ksks             Aleksand a Ss  sov c    CERAK (BEOGRAD       CERAK (BEOGRAD      1111
11  gelenmm          Aleksandra Sre          BEOGRAD              BEOGRAD             1001
12  obradovic        Vlado Bodan             BEOGRAD              BEOGRAD             1111
13  Aleksandra s     V ad m s Ss  ksmd       LEAKSOVO, SIPENE     LEAKSOVO, SIPENE    8811
14  Ivanka           Ivan Todor              BEOGRAD              BEOGRAD             1111
```

```
Najbolja na mesu

Kb  Prezuleim    Prez ne    Ime                 Grad                 Vreme

1   vesna        Vesna  ic  v                   LIKASOVI, LEPENE     1111 88 20
2   vesn         Vesna  ovi  m                  BEOGRAD              1111 88 58
3   gavri        M  ela  Ma  ela                BEOGRAD              88   77 80
4   lemma        Jan  mikmatkov                 BEOGRAD              818  78 81
5   skrel        Ilij  o  rre  m               LIKASOVI, SIPENE     1111 88 12
6   slavni       Pe  ar  Ra  ković             BEOGRAD              111  81 12
7   darmmm       Dusko  Ilas  i                BEOGRAD              881  81 51
8   ivi          Ivi  ni                        BEOGRAD              1111 81 11
9   bojjmy       Bojan  Ss  ov  s              NOVI BEOGRAD         1111 81 84
10  Ndidic       Ivann  D  ko                   BEOGRAD              1111 81 11
11  golmcsss     Aleksandra Sevi  l            BEOGRAD              111  81 11
12  Ss           Aleksandra Ss  ić             CERAK (BEOGRAD)      1111 81 11
13  dgsdic       Dragan  Ra  ić                 BEOGRAD              1111 81 11
14  Ajsvg        Miljan  J  Ranov               BEOGRAD              111  81 08
15  Mihajlo      Vesna  Ra  ić  ev  i          BEOGRAD              1111 81 11
16  Gdvsdick     Ji  im  S  ima                 BEOGRAD              111  81 11
17  Ndid         Ana  D  ić  ima                LIKASOVI, CYPENE     1111 81 11
18  bolj         Boljo  Ra  ić                  NOVI BEOGRAD         1111 81 18
```

```
Najveće poruka s BOG mrežne adre

Kb  Prezuleim    Ime i me    ne                Grad                 Veličina

1   Ssin         Aleksandr  Ss                 BEOGRAD              1111
2   ggsdic       Bodan Ra  ić                  NOVI BEOGRAD         1117
3   kalinda      Bojan Kalinda                 BEOGRAD              1111
4   mssdamic     Blagoslav Ss  ić              BEOGRAD              1111
5   adicjan      Bojan Sedeljic  ić            ZEMUNSKIS            1111
6   madimov      Milan Adamov                  BEOGRAD              1111
7   pdjukovac    Bodan Stevov                  BEOGRAD              1111
8   boljin       Jovan Boljin                  BEOGRAD              1111
9   bmidmy       Ilija Cga                     STOPOVICA            1111
10  BMMMAD       Bodko Stanić                  BEOGRAD              1111
11  Ianil        Boljko Lazić                  BEOGRAD              1111
12  bmjl         Boljin Vujadin                BEOGRAD              1111
13  Ilianovvic   Ivan Lazimm  ić               NOVI BEOGRAD         1111
14  bmidi        Bodan Ra  ić  ić               BEOGRAD              1111
15  bmmdi        Vlajimir Sekulović            BEOGRAD              1111
16  Ivanovic     Petar Ivanović                BEOGRAD              1111
17  Ivanillanvic Ivan Lazimm  ić               NOVI BEOGRAD         1111
18  bmidi        Bodan Ra  ić  ić               BEOGRAD              1111
```

```
Najviše poslatih odgovora

Kb  Prezuleim    Prezime i ime               Grad                 Kb

1   bmvv         Bora Slivanović             BEOGRAD              111111
2   fajlja       Dijana Petrović             NOVI BEOGRAD         111111
3   drnkmm       Nenjan Ratković             BEOGRAD              111111
4   boljaja      Jovan Boljić                BEOGRAD              111111
5   nbmdimvic    Milan Radović               BEOGRAD              111111
6   dnrmmm       Ranka Stanilić              BEOGRAD              111111
7   jlimmm       Andrej Vladić               BEOGRAD              111111
8   jjmmlin      Srdan Pantić                NOVI BEOGRAD         111111
9   pjmmlin      Pavle Petrović              BEOGRAD              111111
10  bmmdi        Bojan Radić                 BEOGRAD              111111
11  bmjmmy       Bojan Pavković              NOVI BEOGRAD         111111
12  Ssstilic     Jadr Ratić                  BEOGRAD              111111
13  Ilmvbk       Radovan Berda               BEOGRAD              111111
14  bmmdi        Milan Radić                 BEOGRAD              111111
15  lbmvic       Berić Slić                  BEOGRAD              111111
16  Ilmvbkmic    Berić Radović               BEOGRAD              111111
17  skmil        Ivan Skerl                  BEOGRAD              111111
```

Ko je ko na Sezamu. Najaktivniji korisnici svih vremena

tvorimo direktorijum PASCAL u koji biemo upisivali sve ove korisne programe

Na kraju pomenimo i dva problema koja nismo uspeli da rešimo – „resetovanje" SHARE-a (ili makar njegovo efikasno uklanjanje iz memorije) i kako ograničiti dužinu MEMO polja u *Clipper*-u. Možda vi imate neku ideju?

PC.USER

Konferencija PC USER bavi se uslužnim programima za PC računara – editorima, tekst procesorima programima za crtanje, projektovanje matematičke proračune. Najviše poruka odnose se na obradu tek sta koja je, očito preovladujuća potreba mnogih vlasnika računara. Tokom protekle godine detaljno smo se bavili poređenjem raznih editora. *QEdit, Multi Edit, Brief* korisnika je naročito zanimala mogućnost da jedan editor da određene mere, emulira neki drugi a mnoštvo poruke odnosilo se i na mogućnost obrade dugačkih (ali zasta dugačkih bar pa megabajte) A-SCII tekstova.

Dosta pažnje posvetili smo tekst procesorima i programima za prelom teksta. „Ht" su bili *WordPerfect 5.0* (pre toga paket *True Type Fonts for DOS* prilagođen *WordPerfect-u* 5.1), beta verzija *WinWord* a 6.0. *AmiPro* i naročito *Ventura for Windows*. Naučili smo da konvertujemo štitnicu u ćiriicu na nekoliko načina rešili problem mislog slova č i što je svakako najvažnije saviadali „oglecisko štampanje". Milan Velimirović (*mivel*) je napisao drajver ALICE SYS koji omogućava „kontrolisht" iz *Ventura for Windows* i drugih *Windows* (i DOS) programa.

Aktivna je bila i tema *cad* – oeim „večitog" pitanja pronalaženje emulatora pomoću kog bi se *AutoDesk* ovi programi korsili na *malinama* bez koprocesora (temu smo verujemo detaljno obradili) često smo diskutovali o štampanju linija raznih debljina na laserskim štampačima i o *AutoLISP*-u

Početkom godina konferencija PC USER dobila je i temu zaštite koje se kako joj i ima kaže bavi kako hardverskim tako i softverskim zaštitama programa. Bilo je dosta zanimljive diskusije na tu temu – savršene zaštita teško da će ikada biti pronađena, ali su smišljeni razni načini da se heksima posao bar malo otešа.

PC.UTIL -

Konferencija PC UTIL posvećena je uslužnim programima sa PC računara – tu se diskutuja o arhiviranja *cache* programima, upravljanju memorijom, virusima konverzijama raznih formata i sličnim temama. Tokom ove godina najviše poruka pristglo je u temu *arhiveri*. Ništa naročito – posle dugog čekanja pojavio se PKZIP 2.04 e onda kao odgovor na njega, i ARJ 2.41. Detaljno smo upoređili sve moguće aspekta od stepena kompresije brzine pouzdanosti i fleksibilnbilo do raznih gotovo nevažnih sitnica. „Ueput" ano rešavali probleme koje su pojedine verzije arhivera donosile na specifičnim konfiguracijama, naročito kada se radi o PKZIP-ovoj DPMI podršci

Druga „hit tema" konferencije PC UTIL bio *Norton Commander* – najpre amo diskutovali o njegovom konfigurisanju i napravili tabelu dokumentovanih i nedokumentovanih komandi, onda su neki korsnici pravili „zakrpe" u samom NC-u da bi ga što bolja prilagodili program svojim potrebama a onda se konačno pojavila nova verzija 4.0. Najpre je izazvala opšte oduševljanje, a zatim doneki i izvesno razočarenje pošto se pokazalo da nije sve baš onakvo kako u reklamama piše

Tema *memory mgr* donela je uglavnom diskusiju o QEMM-u 7.0 u drugoj polovini godine – rešili amo razne probleme vezana se njegovu instalaciju i čudi se kilobajtima kojima „ir ničega" odpunjava osnovnu memoriju. Pa ipak svaka diskusija o QEMM-u donese i raspravu o tome treba li uopšte investirati novac i vreme u njega kada već DOS 6.0 obezbeduje slične usluge? Slično pitanje postavlja se i u vezi *Stacker*-a o kome amo često diskutovali u temi *disk.soft* – ima li on smisla kada postoji „oficijalni" DBLSPACE? Obe teorije imaju svoje pristalice, a neki korisnici su se deklarisali kao „podajsni" – koriste DBLSPACE ali i QEMM!

U temi *virusi* na sreću nije bilo previše posla – jedini stvarno opasan napad bio je „falsifikovani" PKZIP 2.0, zapravo trojanac koji može da naprayi prilično štetu bilo ga je po raznim odmacim BBS-ovima sračunat na i na Sezamu. Ostatak poruka u ovoj temi uglavnom je posvećen programima za borbu protiv virusa – opisivali amo svaku od novih verzija SCAN-a a pažnju je izazvao program *CheckOut*

U ostalim temama bilo je reči o transformaciji slika u boji u nijanse sivog kovertovanju iz jednog grafičkog formata u drugi mogućnosti da se kupaćhat diskalibrisanoti od maksimuma korišćenjem specijalnih programa kao što je FDFORMAT programu PCL koji neverovatno efikasno štiti računar od neovlašćenih ruku, ali ga se teško otarasiti .. Povremeno se javljala i diskusija o „starim dobrim vremenima" detaljno amo

proučili *shareware* simulatore *Spectrum* i *Commodore 64* i razmotrili mogućnost prebacivanja podataka sa njih (i drugih osmobitnih računara) na PC

ORKA

ORKA je Opština Računarska KonferencijA – dobro mesto za diskusiju o kompjuternoj koji nisu PC *Amiga* ili *Atari* dakle koji nemaju svoju posebnu konferenciju (krajem godine otvorena je tema mac posvećena vlasnicima računara *Apple Macintosh*) Tema broj jedan ove konferencije bila su takmičenje mladih programera – detaljno smo pratili gradsko (u „lažoj kategoriji" od prvih 10 mesta čak 6 su osvojili Sezamovci) republičko i Savezno (sva trojica pobednika konkursa Sezama) takmičenje srednjoškolaca, saznali kako se biraju zadaci za Olimpijadu kako se formira komisija za pregled i kako teče samo pregledanje odnosno bodovanje *Posla* sth h penjerija Jugoslovenska ekipa je kako smo pisali u prošlim „Računarima" učestvovala na Olimpijadi (održuje god zastitvom UN) i postigla i zuzetan uspah Uroš Medić je devojče zlatnu a Vladimir Petrović (*vpetrovic*) bronzanu medalju

Zbog velikog broja poruka na temu takmičenja otvorena je i posebna tema *obrazovanje* koje je sredinom godine primila ogromnan broj poruka. Najpre su kritikovani školski programi nastave računarstva i rogobatni prevodi stranih reči zatim se diskusija proširila na letnju računarsku udžbenika za osnovne i srednje škole a nau pošteđeni ni profesori Kao što to obično bva diskusija je otišla „u širinu" i bila na momentai veoma burne pošle se od nastave programiranja ubrzo došlo i do nastave drugih predmeta a onda smo videli kako to sva izgleda u stranim školama posebno Američkim gde je veći deo predmeta izborna.

Dosta poruka akglo je i u temu *electron* koja se bavi praktičnom elektronikom mnogo su se tražili podaci za razne tranzistora i integrisana kole nadene su adekvatne zamene za neke delova koje ja kod nas jako teško naći dobili emo *cross* asemblere i simulatore za i anze mikrokontrolere. U temi *dlesprl* najviše smo diskutovali o *Sharp*-ovim malim kompjuterima a pažlfkvo je istražen i novi HP4HGX naslednik HP4IISX. U temi *zakoni* bilo je dosta reči o domaćim i inostranim propisima, a najviše nedoumice izazvala su pitanje o carinjenju računarske oprama

Najposećenija tema konferencije ORKA *igre* početkom godine je prerasla u posebnu konferenciju

IGRE

Krajem februara otvorena konferencija IGRE nailta je na izuzetan prjem kod korisnika Sezama – veliki broj poruka se vrednim informacijama pokazuje be i *gre*, što as liče solidnog broja korisnike računara ne guba na značaju

Najpopularnije igre pokazuje se, nieu više arkad na i druga „pucačina" iitolmena tema je primila sesvim umeren broj poruka, a arkadna igra godine verovatno je *Pirates* Intenesovanje za avanture je mnogo veće teme godine svakako je bio drugi nastavak poznate igre *Prince of Persia: Shadow and the Flame*. U relativno kratkom roku uspešno su savladena sve prepreke uniišten zli vezir *Jaffar* i osvojena naklonost lepe princeze Otkrivan je i „bajt lične prirode" koji omogućava nestrpljivim igračima da bez mnogo muke pogledaju i odigraju sva nivoe Povoljnih raskcija izazvala je pojava igre *Flashback* koja je sličnog tipa kao i *Prince* a po mišljenju mnogih ge i prevazilazi Dosta se govorilo o igri KGB e ljubitelji avanture su se oprobali u ulozi spotakara na Divljem Zapadu pirata u južnim morima, Šerloke Holmsa detektive u svemiru Godine nije prošla bez domaćih avantura inspirisanih Sezamom najpre PAUS Srđena Veljkovića (*debeljlencije*) i Darka Milutonovića (*darco*) kojr predstavlje odgovor na SUPI (Sezam U Praksi) e onda i SUPI 4 0 Stanislava Šokorca (*sstokorac*) kao „utuk na utuk"

Zona zabrane letenja nije važila iznad teme *simulacije* – naročito su popularne borbana simulacija lete koje se igraju preko modema. U odgovarajućoj temi organizovano je takmičenja – u finalnoj borbi Marko Knezović (*kenzar*) je savladao Stanislava Šokorca I iako zvanično postao najbolji pilot Sezama. Kada već govorimo o igrama preko modema na treba zaboraviti novu verziju domaćeg Tatrisa (ZBTET 3 0 Dragana Zakića (*dzakic*) i Dejane Rudića (*bestf*) koja je dobila i mogućnost međemskog takmičenja!

Tema *strategija* je bila u znaku na bali tako nova i *gre Civilization* – konkurencije je bila velika tako be su atvorene mnoga civilizacije a neke od najnaprednijih među njima su već savladale i kosmička letove te naselile susedna planeta i zvezde Igrači koji pokušavaju da u igri *Civilisation* stvore što bolju i veću civilizaciju su zahvaljujuć Srđanu Panšiću (*spantic*) dobili par programa koji omogućavaju lakše igranje kao i program uz koji je moguća igra vše igrača.

Otvorena je i tema *frp* posvećena orama koji vole *Fantasy Role Play* bilo pomoću računare bilo „klasičnim" sredstvima Najveće intenesovanje vladalo je za treći nastavak popularne igre *Eye of Beholder*

FORUM

FORUM je bazično politička konferencije svakako najbogatija porukama u čitavoj istorij Sezama – već je pri kraju njen deveti „tom" (svaki FORUM po 5000 poruka) Uzbudljivo vreima u koma živimo daja dovoljno povoda da se „ukrste" razni politički stavovi i pogledi na budućnost Neki od povoda koji su izazvali veoma burne diskusije tokom protekle godina su razuftaš izbore iz decembra 1992 odgovornost vođećih opozicionih političara za neuspeh na izborima bekstvo Jezdimire Vasiljevića iz zemlja smena Dobrice Ćosića hapšanje Vuka I Danice Drašković rasprava o poverenju republičkoj vladi I raspuštanje Skupštine Često su čisto politčke rasprava gubile ne intenzitetu i bvale potianuta rasprevame o dramatičnim skokovima cena I kursa marke nestašlici roba lošem privozu malim platama i drugim svakodnevnim teškoćama

Kada se radi o svetskoj politici diskutovali smo a američkim izborima Ujedinjenim Nacijame i njihovom telima koje su poslednjih godina po prvi put u istorij počela ozbiljno da rada na žalost često na našu štetu o burnim događejima u Moskvi Ipak pokazalo se de je interesovanje za svetsku politiku znatno manje nago ranijih godina

Što se napolitičkih tema FORUM-a tiče imali smo jednu „erupciju" poruka u temi *devojke* „krivac" za nju je Aleksandre Kostić (*princess*) koja je porukom od samo šetri znaka („alo?") inicijala prilv od nekoliko stotina poruka dnevno Trebalo je videti pa verovati

Veliko intenesovanje izazvala je novootvorena tema *isaljenje* gde su korsnici razmenjivali nformacije o načinima da se emigrira kao i o potrebnim uslovima se to. Od posebne pomoći bili su oni koji su se već iselili (ili se i dalje javljaju u granicama mogućnosti) jer se iz njihovih poruka mogto saznati mnogo konkretnih pojedinosti o problemima na koje su naišli pćito su se odlučili na taj korak.

CIVILIZACIJA

Konferencija CIVILIZACIJA je se 11147 poruka druga najposećenija konferencija na Sezamu! To je svakako prijatno iznenađenja jer smo tokom prvih godina rada elaki utisak da je intenesovanje korsnika za slikarstvo muziku filozofiju religiju i slične teme sesvim umereno Možda je za to zaslužna i uzavrela politička scena očito zamoreni svakodnevnjem diskutanti su se vile okrenuli temama koje su bliže umetnosti Najviše poruka su ipak „pmnle" teme namenjena poleminci – *ičiri.stev* ja doneo obilje argumenata za i protiv abortusa, uvođenja veronauke u škole smrtne kazne gde prestaje umetnost i počnje kič Polemika je povremeno „zahvatale" i temu *muzika* a u oktobru se proširila i na temu *književnost* – dnevno je pristizalo po nekoliko desetina pa i stotina poruka o odnosu ruskih klasičnih pisaca (tzv levi ruski pisci) sa jedne i autora popularn ph književnosti druge strane

Bilo je i zanimljive diskusije o odnosu ega superega i etonega i o logičkim paradoksima u temi *filozofija* gregorijanskom i julijanskom kalendaru u temi *religija* ratovanju cara Iraklija na dva fronta i potonjoj propasti Vizantije u temi *istorija* muzeju Salvadora Dalije u Figueres (tema *umetnost*) Oprostili smo se od Desanke Maksimović sešli sa Miloša Crnjanskog nabavili (ra če teško dostupnu) doktorsku disertaciju Ive Andrića a bilo je i vredih autorskih priloga u temama *književnost* i *ahromi*

Lakšem snalaženju u konferenciji CIVILIZACIJA znatno pomaže grupa korsnika koji svakog meseca iz dvejaju najznačajnije i najinteresarnije poruke ako nemati vremena da pratita čitavu konferenciju, pročitajta barem njen *digest*!

VICEVI

Ako se FORUM i CIVILIZACIJA „otimaju" za mesto konferencije sa najviše poruka o konferenciji koja se najviše čita nema dileme VICEVI Trenutno se u dva „jpo" toma ove konferencije nalazi oko 8.500 poruka sa, verujemo preko deset hiljada viceva – ako i nije najveća na svetu ovo je sigurno najveća zbirka viceva u Jugoslaviji Tema su uvek vrlo raznolika od nezbežnih Muje I Hase preko Đetića, Lala mešog Pance Cigane izbeglica, životinja, dnevne politke Tokom 1993 godina detaljno smo „procešljali" zbirku viceva o plavušama (sakupilo ih sa tačno 489 recimo „Zašto plavuša voli kabrioleT? Više prostore za noge" ili „Kako se naziva plavuša se dve moždane ćelije? Trudnica") profesorske biseire iz raznih škola a zapaženo je i mnogo viceva i „poslovica" inspirisanih svakodnevnim nevoljama naročito u prevozu Dobili smo i nekoliko humorističkih tekstova iz domaćih (Indeksov radio pozorišta) i stranih (serija *Monty Python's Flying Circus*) izvora. Dosta priloga stiglo je i u temu *bez veze* – kruna niza viceva tipa „Idu ulicom ženska I dve frajera pa je jedan muva a drugi je pčela" je nekto što bi teško i izumeli čitaoci trlo koj časopisa osim nešeg „a uiici dve babe jedna ida a druga SCSI"

Za kraj smo ostavil jedan „olo tajmer" koji se bavi seda već gotovo zaboravljenom Olimpijadom u Sarajevu – izazvao je brdo smeha i naravno bio izabran za vic meseca. Elem na svečanom otvaranju Olimpijada se kao što i pristoji okupio sian narod sva mesta zauzeta Konferansije počnje „Olimpijske igre u Sarajevu otvorće drug Hamdije" Iz publike se čuje mlakušan aplauz Drug Hamdija izlaz staje pred mikrfon vadi papir I čita „Drugarice i drugovi " Iz publike se čuje slabašen aplauz „OOOOI" Iz publike sa čuje snažno „OOOO" „OOOOOOOOI" iz publike oduševljeno ceo stadion u jeden glas „OOOOOOOOI" „OOOOOOOOOOOOOOOOI" U publici delirijum saku vene, skaču prema govorniku leti kape i rublje ekstaza „OOOOOOOOOOOOOOOOOOOOOOOOI" Konferansije šapuće drugu Hamdiji „Čitaj dalje to su ba olimpijski krugovi"

MALI.OGLASI

Iako na donosi tekstove od trajne vrednosti konferencija MALI OGLASI je rekako važno mesto svakog meseca kroz nju proda preko 1000 ponude među kojima se zbilja nalazi sva I svašta „od igle do lokomotive" Najviše je prirodno onih koji prodaju matične ploče čiekove modeme monitora i drugu polovnu kompjutersku opremu ali je dosta i „nekompjutersleh" ogiasa u kojima se prodaju ll manjaju automobili stanovi nameštaj *hi fi* komponente Čak i kućni ljubmci imaju „svoju" temu! Teško je naravno procenti šta se od ponuđenog zaista i proda (dogovori se uglavnom odvijaju preko privatne pošte) eli ima se utisak da dosta roba nade put do novog vlasnika – povremeno su se prodavale i neka stvari za koja bi malo ko verovao da će nakoga interesovati Sezamovi MALI OGLASI su teko postali izuzetno atraktvna (i potpuno besplatna) berza ponude i potražnja i postepeno prerastaju u jednu od najaktivnijih Sezamovih konferencija Ponude su naravno ograničene na nekomercijalne za ona koji prodaju „na veliko" predviđena je konferencija O-GLASI u kojoj se ponuda drugačija tretraju i plaćaju

Neumorni Sezam

Iako smo ovaj tekst opterećli raznim tabelama, grafikonima, „pitama" i ostalim brojevima koji će možda opseti dimenzije koje ja nas BBS za sve čelu godine dostigao na samom Sezamu ima mnogo toga što nije lako kvantifikovati Možda je to privatna poruka od prijatelјice čestoka svađa u grupi sexy posle 30 minuta dobijen odgovor na pregmarsko pitanja koja vas ja danima mučilo program koji je baš stigao u diskretnijim I koji će tako zgodno primaći u posiu podatak o kursevima marke 30-tih i 50-tih godina ovog veka opis nakog filma koji će vas naterati da iknete do video kluba, Ann ulazak u *chat* i višebojno (od njene strane zabranjeno) *hil* se petnaestak testatura „trobip" minut pre isteka vremena i „injekcija" od 15 minuta gotovo bismo rekli be je Sezam postao čitav jedan svet, i to svet koji povremeno predstavlje divan odmor od sumornog sveta u kome provodimo ostalih 1360 minuta svakog dana Svet koji je tu kada god nam je potrebam i koji će ma pozdravib svojim mimom CONNECT 2400 kada konačno odlučim da predam ovaj tekst. Može li se uopšte poželeti nešto lepše? Svakako ba može da ma kade budem slao tekst o njegovom petom rođendanu pozdrave porukom CONNECT 14400!

SezamFile

Jovan Bulajić

Udarni noviteti u Sezamovim direktorijumima u novembru bili su MS-DOS 6.2 i QEMM 7.03. Izašla je i verzija 5.0 4DOS-a, a veliku pažnju privukao je program za koga njegovi autori kažu da primenom nekih specijalnih tehnika može značajno ubrzati IDE hard diskove. Stigle su i nove verzije *Windows 3.1* drajvera za SVGA grafičke karte, kao i sveži drajveri za Microsoft i Genius miševe.

Udarni noviteti u Sezamovim direktorijumima u oktobru bio je MS DOS 6.2 a vrlo tražen bio je i QEMM 7.03. Ovo naravno ne znači da se na Sezamu mogu naći i komercijalni programi | već je reč o zakrpama (patches) koje se slobodno distribuiraju, i za čiju je instalaciju potrebno imati stariju verziju originalnih programa (MS-DOS 6.0 ili QEMM 7.0x)

U vrhu liste najtraženijih programa u novembru su, uz novi MS-DOS i QEMM biti, kao i obično *Scan* i *Clean* nova verzija *List*-a program koji ubrzava rad sa IDE hard diskovima i tekst se objašnjavam interne strukture **TBrowse** objekata za *Clipper* programere. U oktobru na vrhu liste bila i Microsoftova kolekcija novih *Windows 3.1* drajvera sa SVGA grafička karte

Tabela 1 – TOP 10 datoteka – novembar 1993

	direktorijum	datoteka	opis	
1.	\IBMPC\VIRUS	scanv109	a01	McAfee SCAN ß.20v108 – virus detektor
2.	\IBMPC\VIRUS	clean109	a01	McAfee Clean ß 20v108 – uklanja viruse koje SCAN nađe
3.	\IBMPC\SYSTEM	madoe82	a01	MS-DOS 6.2 – Step-Up update sa verzije ß 0
4.	\IBMPC\SYSTEM	702to703	zip	Upgrade sa QEMM-a 7.02 na 7.03
5.	\IBMPC\UTILITY	list90c	zip	List v9 0c – najpoznatiji program sa pregled datoteka
6.	\IBMPC\DISKUTIL	dqwik110	zip	DiskQwik v1 1 – ubrzava rad AT/IDE hard diskova
7.	\IBMPC\CLIPPER	tbmore	zip	Objašnjenja interne strukture TBrowse objekata
8.	\WINDOWS\WUTIL	winzip50	zip	WinZip 5 0 – shell za rad sa ARJ/ZIP/LZH/ARC arhivama
9.	\IBMPC\VIRUS	vsh109	zip	VShield ß 20v108 – rezidentni virus detektor (SCAN)
10.	\IBMPC\SYSTEM	4dos5new	zip	Šta je novo i izmenjeno u 4DOS verziji 5 0

Noviteti

U direktorijum **VIRUS** stigao je kompletan paket novih verzija svih mogućih alata za borbu protiv virusa kompanije *McAfee – Scan, Clean i VShield ß 20v109*, bazu podataka Patricia Hoffman sa detaljnim opisom svih poznatih virusa (*VSumX*) *NetShield 1.55v109* (zaštita od virusa za *Novell Netware 3.1x*) i *Windows* verzije v108 *Scan*-a (u WUTIL direktorijumu) izašla je i verzija 2.10 (november '93) antivirusnog programa *F-Prot*, o kome smo pisali u prošlom broju

Od poznatijih uslužnih programa, tokom prethodna dva meseca nove verzije dobili su i *TSR Utilities* ver zija 3 5 (paket koji sadrži pozvate programčiće za rad se različitim programima – *Mark Release*, *MapMem* i al) *Shez v9 5* (i dalje standardno izlazi neka "nova" verzija svakog meseca), *DiskDupe Trial 4 0ße* shareware verzija programa sa kopiranje diskete, *List 9 0c* najpoznatiji program za pregled (browse) datoteka (bio je i jedan od najtražanijih programa u novembru), *PC-File v7 0* sistem za rad sa DBF bazama podataka, *Q387 v2.8* softverski emulator 387 koprocesora, *WinZip 5.0 Windows shell* za rad sa arhivama, *DFlat++ 3.0* C++ biblioteka klasa za korisnički interfejs i, na kraju, najnovije verzije *4DOS-a 5.0*

Što se kvantitativnih parametara tiče ukupno su tokom oktobra stigle "sveže" 63 datoteke ukupne dužine 9.5 megabajta, dok je u novembru bilo 107 novih datoteka dužine 14.3 Mb

Od novih programa najveću pažnju privlači je *DiskQwik* Reč je o *device* drajver-u dizajniranom sa AT/IDE hard diskove koji aktivira njihov tzv "multiple sector block transfer mode" Autori programa kažu da sa na ovaj način brzina prenosa sa diska (*transfer rate*) može povećati za 45%, i to na skoro svim IDE diskovima i konfiguracijama (u paketu se dobija i program sa testiranje koji odmah kaže da li vaš disk podržave tu mogućnost ili ne)

Shareware verzija koja se nalazi na Sezamu ograničena je na dva sektora po bloku i na jedan drajv- ali se i njome mogu postići dobici od nekih 10 do 15%. Korienici koji su "registrovali" program na 32 sektora po bloku kažu da *Core Test* pokazuje poboljšanje od čak 100%, a da zabljnuji i rasinji testovi izmere ubrzanja od oko 30% Inače vam skoćamo požnju da na verujete bal pravile rezultatima koji pokazuju *Core Test* i slični programi – kada smo nedavno instalirali o bični *Local Bus* IDE disk kontroler *Core* je pokazao da su transfer rate i *Performance Index* porasli više od dva puta (transfer rate je bio neverovatnih 3.500 kb/s, bez lkakvog cache-a), dok ja realno ubrzanje ipak bilo umerenih 30%

Iskustva sa primenom *DiskQwik*-a su, još uvek, nedovoljna da bismo sa sigurnošću rekli da li je reč zaista o odličnoj stvari ili možda o prevari. Ipak, nama vas na kolita da ga pokupite sa Sezama i probate (**\ibmpc\utility\dqwik110.zip**)

DOS

Microsoft je izdao **MS-DOS 6.2** *Step-Up update* Reč je o nekoj vrsti *patch*-a koji od zvaničnе verzije MS-DOS-a ß 0 pravi verziju 6.2 Kažemo zvaničnе verzije jer je u našim krajevima dosta rašlrena "novinarska" verzija, koju je Microsoft delio časopisima na recenziju pre zvaničnog izlaska novog DOS-a. Na žalost sa tom verzijom koja nosi datum 22. februar 1993 ovaj Update neće raditi već je potrebna verzija od 10 marta 1993

To nije kraj potencijalnim problemima koji vas mogu čekati ukoliko se odlučite da uradite update na MS-DOS 6.2 Ukoliko ste slučajno obrisali DBLSPACE.BIN (a sko vam *Double Space* na treba, verovatno ste ga obrisali), moraćete nekako da ga vratite jer bez njega instalacioni program za *update* neće da radi. Naviše smo već da Microsoft nije u stanju da napravi poštan operativan sistem ali je tragično to što mali Billy ne može da napravi ni najobičniju normalnu zakrpu za ieti

Sami noviteti u MS-DOS verziji ß 2 sigurno ne opravdavaju toliki skok u oznaci verzija već je reč, pre svega, o marketinškom potezu da ti se bilo ispred IBM-a, koji je takođe napravio novu (bolju) verziju DOS-a Poboljšane je pouzdanost programa *DoubleSpace* i *SmartDrive* a zgodna novina u MS-DOS-u 6 0 de se config sys može izvršavati linlju po linlju. sada je proširena i ne autoexec.bat. "Revolucionarna" novina je to što konačno posle ko zna koliko godina i verzija DOS-a *DiskCopy* može da kopira diskete u jednom prolazu.

Jedina potpuno nova stvar u ovoj verziji DOS-a je utility **ScanDisk**, koji detektuje dijagnostikuje i popravlja greške na hard diskovima, kao što su *lost clusters*, *crosslinked datoteke* ili fizičke greške na disku. I ovo je stvar koju su još davno mnogi drugi napravili bolje npr *Norton Disk Doctor*

Naravno ako koristite MS-DOS 6.0 ipak vam sa veljamo da skinete sa Sezama update (koji je do nas stigao zahvaljujući kolegama sa imtel BBS-a) koji je ukupno dugačak 1.4 megabajta, a nalazi se u direktorijumu **\ibmpc\system** u datotekama **madoe82.a01** do **.a07**

4DOS 5.0 je najnovija verzija popularnog komandnog procesora, nezamenljiva zamena za **command.com** Njegovi brojni korisnici već znaju koje sve pogodnosti pruža 4DOS a onima koji ga još ne koriste preporučujemo da obavezno pogledaju i upoznaju novu verziju Sa ovom novom verzijom će im prvi korak u upoznavanju bil znatno lakši – dok je ranije 4DOS bio poznat po svojoj enormno velikoj dokumentaciji (tekst datoteka od preko jednog mababajta) sada je to sveden na razumnih 220 kilobajta, koje i nije toliki problem pročitati Zato je njegov help (u *HyperText* obliku možete ga uvek pozvati pritiskom na F1 iz 4DOS prompta) dvostruko uvećan ali na žalost, u njemu se čeka nego ranije sa dodatne informacije upućuje na *Reference Guide* (koji se dobija registracijom)

Same spisak stvari koje su u *4DOS 5.0* promenjene u odnosu na poslednju verziju 4.0x dugačak je preko 20 Kb Na primer sada sve komande se rad se datotekama mogu primati i datumski opseg pobolišanu je editovanje komandne linije i *history* list, a da biste imali lepe DIR-a u raznim bojama više ne morate imati instalran ANSI drajver (što znači da možete da ušteđite nekoliko Kb koliko on zauzima).

Značajan je i novi *BatComp utility* koji kompresuje (i lekovemeno kriptuje) 4DOS-ove *batch* datoteke (.BTM) Za razliku od MS-DOS-a i njegovih BAT-ova, 4DOS ima moćan jezik za pisanje *batch* programa u kome su napisane mnoge jako korisne alatke – na primer razni *archiver shell*-ovi *file managerl* i tome slično U verziji 5 0 taj jezik je dodatno pobolišan (npr dodata su DO WHILE i DO UNTIL petlje) a *BatComp* će dobro doći autorima tih nezavidnih alatki I 4DOS se nalazi u **\ibmpc\system** direktorijumu u datotekama **4dos50a.a01** i **.a02** (program i dokumentacija) i **4dos50b.a01** i **.a02** (help)

Drajveri

Microsoft je izdao najnoviju kolekciju *Windows 3.1* drajvera sa SVGA grafičke karte u 256 boja. Pokrivaju skoro sve ne-akceleratorske grafičke karte sa rezulucijama 640*480 800*600 i 1024*768 u 256 boja se ma šim ili velikim fontovima. Ovi novi drajveri dosta su brži od starih e u nekim slučajevima (npr za Tseng ET 4000) su brži i od *Panasce WinSpeed* drajvera koji su ranije bili poznati kao vrlo ubrzani

Video kartice za koje se nalaze drajveri u ovom paketu su ATI VGA Wonder Cirrus Logic, GAK (077 serija), Paradise Trident (8900c serija), Tseng ET4000, Video Seven i Western Digital VGA. (**\windows\wdrivers\svga256.a01** – **.a02**)

A ako kojim slučajem imate **Cirrus Logic 542x SVGA** kartu i želite da je ubrzate uzmite novu verziju 1.4 *Windows* drajvera za CL 542x iz WDRIVERS direktorijuma. Sudeći po menjnima u *WinTach* teku ubrzanja su neverovatna – u modu 800*600 sa 64K boja novi drajveri su u ukupnom indeksu dvostruko brži od starih (koji su nosili oznaku verzije 1.32). e u 800*600 u 256 boja skoro četiri puta više! Realno ubrzanje je ipak mnogo manje – mora da su napravili driver koji prepoznaje *WinTach* i vara ga) ali je ipak primetno.

Ako želite da obradujete svog Microsoft ili Genius miša, nabavite mu najnovije verzije drajvera sa DOS i *Windows* koje možete naći na Sezamu Oznaka verzija su za Microsoft i kompatibilna miševe ß 01 08 beta (se ista čudna oznaka verzije ali sve radi Ok i ne izgleda kao beta verzija) i za Genius verzija 10 20

U slučaju da je tela vašem mišu Ell Gejts novi drajveri će ga naučiti raznim trikovima – npr pod *Windows*-om će moći da *mause pointer* isače sa jednog kraja ekrana i da se pojavi na drugom može se podešavati osetljivost i ubrzanje nezavisno po obe koordinate (vertikalno i horizontalno) e moguće je napraviti veoseoiie kuror (npr prako četvrtine ekrana)) I *Mouse Manager* sa DOS, takođe, omogućava razna podešavanja.

Ovi drajveri se mogu naći u *\ibmpc\drivers* direk torijumu I to kao datoteke *mouse901.a01* – *.a04* (ukupne dužine skoro 700 Kb) za Microsoft miševe, a **gmous102.zip** za Genius.

INSTITUT ZA NUKLEARNE NAUKE "VINČA"
CENTAR ZA PERMANENTNO OBRAZOVANJE

Telefon: 011 / 683-390, 682-486, 641-155 / 107, 181
Telefax: 011 /682-486

QUATTRO PRO · FRAMEWORK · AGENDA · SPSS / PC+ · ORACLE · FoxBASE
MS WORKS · PARADOX · FREELANCE GRAPHICS · cc Mail · CLARION · MAGIC
MS EXCEL · LOTUS 1-2-3 · SYMPHONY · PRIMAVERA · ACCESS · dBASE
C.A. CLIPPER · FoxPro

TURBO C · C++ · COBOL · PASCAL · FORTRAN 77
WORDPERFECT · MS WORD · WORDPERFECT FOR WINDOWS · WORD FOR WINDOWS · AMI PRO
AutoCAD · AutoLISP · AutoSHADE · 3D STUDIO · SOLID · CorelDRAW! · AEC ARCHITECTURAL
VMS · UNIX · XENIX · OS / 2 · MS DOS · NOVELL · WINDOWS

KURSEVI U OBLASTI PROGRAMIRANJA, PROGRAMSKIH JEZIKA I PRIMENE RAČUNARA

AUTODESK LTD

BORLAND

JEDINI U SRBIJI AUTORIZOVANI OBRAZOVNI CENTAR SOFTVERSKIH FIRMI

Lotus

SCO THE SANTA CRUZ OPERATION

IZDANJA U OBLASTI PROGRAMIRANJA, PROGRAMSKIH JEZIKA I PRIMENE RAČUNARA

01. AutoCAD (verzija 11.0) - konstruisanje i projektovanje pomoću personalnih računara
Autori: Boris Damjanović i Petar Damjanović
02. Uvod u C jezik; Autor: Vladan Vujičić
03. Primena programa SYMPHONY na personalnim računarima; Autor: Dragan Pantić
04. OS/2 vodič za korisnike, Autor: Zorica Jelić
05. VENTURA 4.1; Autor: Dmitar Stevanović
06. FORTRAN 77 - standard sa dopunama za personalne računare, Autori: Vlajko Kocić i Zoran Konstantinović
07. UNIX vodič za korisnike, Autor: Zorica Jelić
08. Primena programa FRAMEWORK III na personalnim računarima; Autor: Dragan Pantić
09. PROGRAMSKI ALATI U MATEMATICI - MathCAD, Grapher, Eureka, Autor: Ante Ćurlin
10. QUATTRO PRO 4, Autor: Dragan Pantić
11. DOS 6.0 UKRATKO, Autor: Dragan Pantić
12. Vodič za VAX/VMS, Autori: Tamaš Kerepeš, Zvonko Orsolić, Saša Matijević
13. Primena programa EXCEL na personalnim računarima, Autor: Dragan Pantić
14. UNIX vodič za programere, Autor: Zorica Jelić
15. WINDOWS 3.1, Autor: Dragan Pantić
16. PRIMAVERA - upravljanje projektima uz pomoć računara; Autori: Jaroslav Urošević i Jelica Draškić Ostojić
17. dBASE III+ priručnik, Autor: Milorad Filipović
18. Osnovi informacionlogije i informacione tehnologije, Autor: Ljubomir Dulović
19. LOTUS 123 (verzije 3.0 i 3.1), Autor: Dragan Pantić
20. dBASE IV priručnik, Autor: Ljubomir Lazić
21. WORDPERFECT (verzija 5.1), Autori: Dragan Pantić i Nada Pantić
22. Programiranje u C.A. CLIPPERu 5.2; Autor: Alempije Veljović
23. FoxPro, Autor: Dušan Časić
24. UVOD U STRUKTURE PODATAKA, Autor: Miroslav Jecković
25. ORACLE (verzija 5) arhRektura i administracija, Autor: Vladimir Milojković
26. WORD ZA WINDOWS, Autori: Dragan Pantić i Neda Pantić
27. MS WORKS upravljanje poslovnim procesima, Autori: Mirjana Nikolić i Miroslav Nikolić
28. PASCAL standard sa dopunama za personalne računare, Autori: Zoran Konstantinović i Slobodan Simić
29. OPERATIVNI SISTEMI - MSDOS 3.30 do 6.0, NDOS 6.0 i 6.01, DR DOS 6.0
Autori: Dragan Cvetković i Radica Cvetković
30. LOTUS 123 (verzija 2.3) i modeli za poslovno odlučivanje, Autori: Mirjana Nikolić i Miroslav Nikolić
31. PC PAINTBRUSH IV PLUS (WINDOWS 3.1 APPLICATION), Autor: Miodrag Belošević
32. UVOD U SISTEMSKO PROGRAMIRANJE III, Autor: Dušan Velašević
33. MS DOS 6.0, Autori: Mario Ratančić i Branislav Tadorović
34. QUATTRO PRO ZA WINDOWS, Autor: Dragan Pantić
35. EXCEL 4.0 - ukratko, Autor: Dragan Pantić
36. NORTON 6.0; Autori: Dragan Cvetković i Radica Cvetković
37. PC TOOLS 6.0, Autori: Dragan Cvetković i Radica Cvetković
38. MICROSOFT OKRUŽENJE - WINDOWS, WORD, EXCEL, WORKS Autori: Mirjana Nikolić i Miroslav Nikolić

DESETA SVEĆICA NA TORTI

Ovim brojem palimo daesetu svačicu na rođandanskoj torti „Računara" – poslednjih dana decembra davna 1983. godina, na avim kioscima u zamiji pojavili au aa „Računari u vašoj kući", specijalno izdanja časopisa „Galaksija" i zametak našeg časopisa. Za tih daaat godina promanlio aa mnogo toga, i to na samo u uvak tako dinamičnoj obiaati računarstva: dogodilo aa da au „Računari" nadživali na samo „Galaksiju", nego i državu u kojoj au počali da izlaza! Oavrnimo se, dakia, na protakiu godinu, koju čamo avakako pamtili po siromaštvu i astronomskoj inflaciji, ali koju ča računarski avet pamtili pre svaga po hardvaru!

Dejan Ristanović

Godina 1993 poseino će se sećati ani koje zanima tržište mikroprocesora. Pojavio se pre svega Pentium novo čudo Intelove tehnologije. Ovaj (kako su ga mnogi nazvali) „superčip" je zasnovan na 0.8 mikrometarskoj 5 volinoj BiCMOS tehnologiji i ima preko 3 miliona tranzistora. Zaokružio je Intelov po svemu izuzetan proizvodni ciklus u kome je tokom poslednjih petnaestak godina nastalo čak devet generacija mikroprocesora, na svaka 44 meseca po jedan. Svaka sledeća generacija se ponosila duplo bržim radom od najstarijih predstavnika prethodne, pa ni Pentium nije izneverio ovo očekivanje: svi testovi pokazuju da je Pentium na 66 MHz dva i više puta brži od 80486 DX2 na istoj frekvenciji. Što je još važnije aplikativni testovi pokazuju da se ovaj odnos očuvava i u mnogim praktičnim primenama, uz očekivane izuzetke koju može da izazove magistrala posredstvom koje se komunicira sa periferijom (ISA, EISA, Micro Channel Local bus itd). Da bi postigao ovu brzinu Intel je morao da se odreknе svoje tradicionalne CISC arhitekture i da zagazi u vode koima su do sada plovili samo konstruktori brojnih RISC procesora. granica od 100 MIPS-a probijena je tek primenom superskalarne arhitekture i paralelnim izvršavanjem dve instrukcije. Intel, naravno nije mogao da se odreknе vertikalne kompatibilnosti – Pentium ii monso da izvršava sve programe napisane za 8086 80286 80386 i 80486 po mogućnosti znatno brže od bilo kog prethodnika – zacrtano je na primer da Pentium bude bar 150 puta brži od svakog 8088! U tome su zaista i uspeli ali na bez plaćanja odredene cene

PROCESORI I MEMORIJA

Ta cena se pre svega svodi na (pre)veliki broj tranzistora: teškoće pn proizvodnji tako složenog sklopa, visok škart i pregrevanje čipa Intel, zapravo, trenutno na može da proizvede ni približno onoliko Pentium čipova koliko je tržištu potrebno pa mnogi proizvođači već dizajniranе ploče i delje drža u laboratorijama prodajući tek jedan ili dva modela a i to vi še zbog prestiža nego zbog zarade. Da bi stvar bile još gora 80486 računan se sve slabije prodaju jer kupci na žele da ulažu u tehnologiju koja je praktično prevazidena. U zbiru iako je Pentium dobio izuzetno dobre kritike i položio brojne testove Intelove akcije sa početkom decembra pale za gotovo 10 procenata. To je po tržišnoj logici koju nije uvek lako razumeti obrilo i akcije drugih proizvođača čipova pa je na primer Motorola „ni krive ni dužne" pala za 3 do 5 procenata. Svi ti pomaci naniže rezultat su privremenih teškoća – ako Intel uspa da stabilizuje proizvodnju Pentium-a reši problem sa njegovim hlađenjem i ponudi nešto niže cene akcija će svakako početi da raste bar dok Pentium na bude ozbiljnije ugrožen od konkurencije!

A i glavna konkurencija je tokom ova godina promovisala svoju uzbanicu: alijansa koju čine IBM Apple i Motorola predstavila je prvi rezultat dvogodišnjeg razvoja – procesor MPC601 nadonačelnika budućа PowerPC porodica. Ambicije „velike trojke su sve samo na skromne – elijansa je canovana da bi sa kako kažu, „prigrabilo i kontrolisalo ključne personalnih računara XXI vela" PowerPC proceson nau „hibridi" RISC i CISC tehnologije i ne oplerišuje ih vertikalna kompatibilnost sa patnaestak godina starijim prethodnicima. To su severnemi RISC procesori koji pretenduju na izuzetnu brzinu i ipak zahtevaju znatno manje tranzistora (MPC601 se sastoji od „samo" 2 5 miliona) manje se grnju i što je posebno važno mnogo manje koštaju. Za sada još nema šira prstupačnih računara zasnovanih na MPC601 i i kada se pojave mučiće ih hronična nestašica softvera. Jer PowerPC je kompatibilan samo sa postojećom IBM-ovom tehnologijom koja je dobila ime POWER što je lepo izabrana skraćenica od *Performance Optimized With Enhanced RISC*. Poiito POWER programa na tržištu ima malo PowerPC računan moraće za prvo vreme da emuliraju PC ili Macintosh kompjutera – u prošlim „Računarima" videli smo da taj zadatak nije nimalo jednostavan i da je većina potrebnih alata još u razvoju. No trošlana alijansa nije u tolikoj žurbi – rešili su da rade temaljno trudeći se da Intel ne ostane kao jedini proizvođač mikroprocesora najvše tehnologije. Sledeća godina će pokazati ko je koliko uspešan u tom mikroprocesorskom ratu

Proteklu godinu obeležio je i jedan veoma neprijatan događaj– požar u fabrici „Sumitomo Chemical" (Niihama, Japan) desetkovao je svetsku proizvodnju materijale zvanog *epoxy cresol resin*. Ova smbie je glavni sastojak plastičnih kućišta u koje se ugrađuje većina savremenih čipova, a „Sumitomo Chemical" je zadovoljavao 60 posto tržišnih potreba za njom – iznenada „imati smoku" ovoga puta se pretvorile u „nemati smolu". Cene DRAM čipova otišle su „u nebesa" – od tridesetak dolara ze megabajt cena je začas skoči la do 100 i više dolara što se itakako odrazilo na cene gotovih sistema prodaju drugog hardvera (pa i softvera). Od požara 4 juta pa do kakve-takve konsolidacije tržišta u septembru kompjuterska industrija trpela je prilične potrese. No baz svakoga se može pa i baz fabrike „Sumitomo Chemical" – snalažljivi Japanci krenuli su u potragu za alternativnim materijalima a pokazalo se i da su neke veće firme imale solidne zalihe „za žiu na trabalo" – organizacija zvana „Semiconductor Industry Association" uspela je da organizuje postojeće zalihe i tako kbiiko-toliko snabde tržište pa su cena DRAM čipova pale na 60 dolara za megabajt. Prave konsolidacije tržišta među tim došle je tak prvih dena decembra kada je „Sumitomo" obnovio proizvodnju – DRAM čipovi u Sjedinjenim Državama sada koštaju četrdesetak dolara po megabajtu sa tendencijom daljeg pada. požar su počeli da zaboravljaju sv oem porodica onog nesrećnika koji je u njemu egoreo. Tranutno nedostaju samo 4-megabitni SIMM-ovi (*Single Inline Memory Modules*) – na njih prelaze mnogi raniji konsnici 1 megabitnih SIMM-ova a proizvođači su se uglavnom koncentrisali na 16-megabitne koji će dominirati tokom sledeći godina

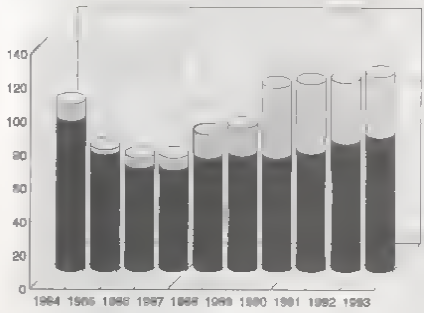
USPEH LOKALNE MAGISTRALE

Procesori rastu iz generacije u generaciju, memorija se i pored ovih neprijatnosti množi eli komunikacija sa spoljnim svetom kao da već duže vreme stagnira ISA magistrala koju koristi velika većina postojećih personalnih računara je pravi „dinosaurus" odgovoran za to što računan rade mnogo sporije nego što bi realno mogli. No ni njeni naslednici nisu sa proslavili – ambiciozno najavljivana EISA magistrala „osvojila" je delić tržišta ali već godinama na uspeva da sa nametne kao stvarni industrijski standard IBM ov Micro Channel prstpeo je više revizija i sada je prilične cenjen, „ali zastupljen gotovo isključivo kod računara iz serije PS/2 koji nikada nisu ni pretendovali da zamene PC klonove renomeranih (i onih drugih) proizvođača koji dominiraju tržištem. Tako je sulom prilika, „ne svoje" dolila arhitektura kojoj su se mnogi u početku smejali – *Local bus* ideje ovog standarda je de standard nija ni potreban – svi relevantni signali sa mikroprocesora dovode se do ekspanzionih portova, pa naka sa periferije sa njima analazi „kako zna i ume" To je u početku uslovilo pojavu različitih međusobno nekompatibilnih grafičkih kartica disk kontrolera i drugih komponenata ali se relativno brzo formirao sada dominirajući VESA standard. Kupovinom matične ploče sa VESA slotovima razmerno ste sigurni da ćete i sada i sledećih godina moći da širite sistem u skladu sa potredama i mogućnostima čak i ako se u međuvremenu čemu se svi nadaju konačno pojavi neki standard za magistralu koji će biti dovoljno dobar da se nametne tržištu

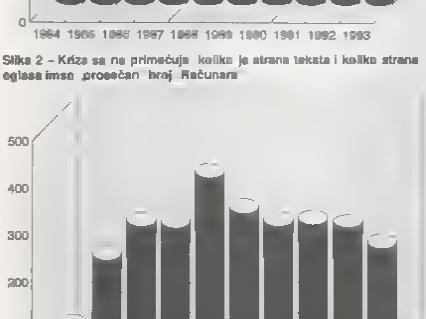
U VESA slotove obično se uključuju video kartice i disk kontroleri. Video kartice su sva češća opremljena posebnim grafičkim mikroprocesorima koji obezbeđuju izuzetno brz rad pod *Windows*-om ili iako kojim drugim operativnim sistemom kome je kartica specijalno prilagođena. A sva to po cani koja je čak i niža od onoga što smo pre samo godinu dana plaili za neku sada tako prazne vrednu *Tseng* ili *Trident* SVGA kartu! Što se disk kontrolera tiče „u modi" su isključivo IDE i SCSI – na MFM RLL pe i ESDI već smo i zaboravili. SCSI kontrolen su obično skupli ali zato imaju razne pradnosti pre svega mogućnost priključenja većeg broja diskova na sistem e zatim i keš od 1..2 4 ili više megabajta na samom kontroleru. Različita su mišljenja o tome da li je bolje investirati u keš kontroler ili proširiti canovnu memoriju računara pa konstiti neki dobar softver za keširanje diska – obično se ne računaru koji će raditi pod DOS-om opredeljujemo za proširenje osnovne memorije a kod mrežnih servera i višekorisničkih operativnih sistema za keš kontroler. Većina kupaca. Ipak, uzima IDE Što se samih diskova tiče cena su im tokom 1993 godine drastično pala: bar kade se radi o modelima kapaciteta ispod 300-350 megabajta. Veći diskovi su i delje skupi pa se tako prilično paradoksalno događa da je primerlio jeftinije kupiti dva diska od po 350 megabajta nego jedan od 700. Verovatna će i cene većih diskova uskoro pasti čemu se svi uvek tako iskreno obraduju – softver je kada se radi o piratoru sve zahtevniji pa iako više nije nikakva retkost da neki programski jezik zauzme 50 ili 70 megabajtaI Paket po paket i disk od 500 megabajta postaje lesani

U ZNAKU OPERATIVNIH SISTEMA

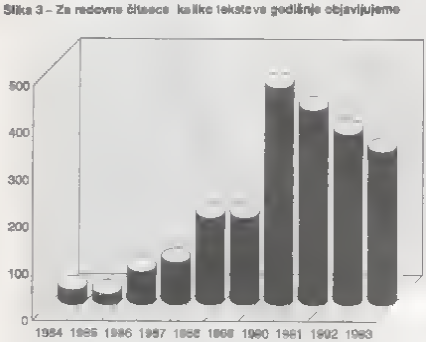
Kada je reč o softveru 1993 godina je svakako prošla u znaku operativnih sistema I to pre svega zaslugom Microsoft-a u martu je pre svega predstavljen MS DOS 6.0 ked tiniaest godine star operativni sistem čiji kojim radi više stotine miliona personalnih računera u svetu DOS 6.0 kao ni prethodna verzija nije donio ništo što u PC svetu nije već viđano – glasna novost je DBLSPACE program koji omogućava da bez ikakve investicije u hardver „udvostručite" slobodan prostor na svojim diskovima. Podaci se naime pri snimanju na disk komprimuju e pri učitavanju raspakuju i to sve bez intervencije korisnika i uz jedva primetno usporenje sistema. Glavno pitanje koje se postavljalo jeste koliko je sve ovo sigurno i može li se deseti deo zbog neke nesrećno zadate naredbe nestanka struje ili daleko bilo bage u programu ostanete bez kompletne particije. Pokazalo se da je DBLSPACE veoma pouzdan – probleme je ako je računar ispravan meio ili ih uopšte nema pa je DBLSPACE stekao simpatije začudujućeg broja vlasnika PC računara. Popularan je i drugi važan novitet DOS-a 6.0 MEMMAKER program koji omogućava lakU i udobnu optimizaciju gornjeg bloka memorije na



Slika 2 – Kriza sa na primećuje kolika je stvarna tekata i kolika strana oglasa imsa „prosečan" broj Računara



Slika 3 – Za redovne čitaoce kolike tekstove godišnje objavljujemo



Slika 4 – Domaće računarsko tržište i dalje je živo. kolike je oglase kilo u „Računarima"

God.	Strojeva	Oktavo	Sof.-oktavo	Tekstova	Veći	Manji
1984	3	500	21	64		17
1985	7	590	30	313		52
1986	11	705	74	286		56
1987	13	816	56	283	178	37
1988	13	1070	185	307	376	97
1989	11	940	199	312	193	59
1990	11	1219	462	389	98	32
1991	10	1181	415	280	173	55
1992	10	1132	364	378	624	61
1993	9	1048	397	204	139	62

Slika 1 – Decenija izražena brojevima osnovni parametri „Računara u periodu 1984 – 1993.

386 (i boljim) kompjuterima i tako čini da granica od 640 K koju DOS nikako da „preskoči" bude sve manje neprijatna. Tu su i razni drugi sitniji novitati dodatne sistemske komande kao i programi koji obezbeđuju veću sigurnost podataka i lakše manipulisanje njima.

Neki domaći korisnici su se pravom primetili da se i bez DOS a 6.0 sasvim lepo živi – *Stacker* je dobar bar koliko i DBLSPACE, QEMM je bolji od MEMMAKER-a *LapLink* od INTERLNK-a. Takve teze međutim samo pokazuju da se ovde do softvera lako dolazi – novac koji u normalnim uslovima treba uložiti da bi se od DOS-a 5.0 napravilo nešto što liči na DOS 6.0 nije bio zanemarljiv pogotovu kada neko treba da nabavi po više desetina ili stotina primeraka svakog od pomenutih paketa za desetak ili sto računara u svojoj firmi. Umesto toga, pisti se licenca za DOS i sve je tu pod nekim „kapom" DBLSPACE MEMMAKER MSBACKUP INTERLNK, neki program defragmentacije diska, antivirus softver. Desetak miliona ljudi kupilo je DOS 6.0 obezbeđujući tako *Microsoft*-u vrlo pristojnu zaradu i novac potreban da, možda već sledeće godine, predstavi DOS 7.0.

„Šesta mladost" MS DOS-a i veliki uspeh njegova nova verzija navela je neke proizvođače softvera da zaboravse na pomisao da je DOS „mrtav" firma WordPerfect Corporation je na primer posle par verzija za *Windows* učinila svojevrsan „korak nazad" u vidu paketa *WordPerfect 6.0 for DOS*. Verovatno su želeli da sa ponove obrata onima koji nemaju mogućnost da drže korak sa generacijama hardvera koje smenjuju jedna drugu – pokazalo se da i na skromnim 386SX i 386 računarima može biti realizovan sasvim solidan WYSIWYG tekst procesor koji radi u grafičkom modu. Mnoge druge firme o takvom koraku na pomišljaju – programira se samo za *Windows* grafički paketi za *Windows*, tekst procesori za *Windows* programski jezici za *Windows* uslužni programi za *Windows* atlasi sveta za *Windows*. *Windows* čak i u primeni na kojima objektivno nije potreban *Microsoft Windows 3.1* je, kratko rečeno iz korena promenio sliku tržišta PC softvera – pomogao je kako malim firmama koje više ne moraju da brinu o prilagođavanju svojih programa specifičnim štampačima i grafičkim karticama, tako i velikim proizvođačima softvera koji su dobili priliku da standardizuju svoj korisnički interfejs. *Windows*-u se zamera sve i svašta – da je spor, da zauzima previše prostora (kada je krajem DBLSPACE pojavio se i vic „udvostručite kapacitet svog diska. Otkucajte DELTREE WIN") da se teško konfiguriše da povremeno „puca" da je tek neuspele i više godine „zakasnela kopija korisničkog interfejsa za *Macintosh*. Na neka od tih zamerki se može ovo ili ono odgovoriti ali ona najveća ostaje. *Windows 3.1* nema pravi *multitasking*, već tek vi šeprogramski rad koji se zasniva na „dobroj volji" aktivnih programa (*cooperative multitasking*) umesto na njihovom smenjivanju pod strogom kontrolom operativnog sistema u skladu sa dodeljenim prioritetima (*preemptive multitasking*). Pa ipak *Windows* je osvojio PC tržište i gotovo zauzeo mesto koje je nekada pripadalo DOS-u donoseći *Microsoft*-u izuzetna priho de. Gejts je rekli bi neki cinici osvetio nešto što je malo kome pošlo za rukom – da monopol jednog svog proizvoda (DOS a) zameni monopolom drugog (*Windows*) e da ipak svako kome je potreban novi proizvod mora pre toga kupiti i stari, pošto *Windows* na radi iako DOS nije instaliran!

WINDOWS NT ILI OS/2

Ma koliko *Windows* bio uspešan i prihvaćen na može se većito od njega živeti pa je tako i Bil Gejts posle nekoliko odlaganja odlučio da prezentira svoj novi operativni sistem u okviru uvodnog predavanja na poznatom sajmu *Comdex* 25 meja ove godine. Pokazalo se na žalost, da promocija na označavaa i početak isporuke – prvi primerci *Windows* NT-a isporučeni su korisnicima tek tokom avgusta, a onima koji su planirali da koriste NT na mrežnim serverima i dalje se samo savetuje da budu strpljivi. Gejts je tokom predavanje rekao da je NT prvi operativni sistem koji kombinuje *client-server* aplikacije sa glavnim poslovnim aplikacijama. „To će" rekao je Gejts „biti ništa manje nego fundamentalna promena načina na koji kompanije formulišu svoje poslovno-računarske zahteve. Sav biznis koji se oslanja na računare sada će moći da se odvalje na istoj platformi u punoj meri uživajući blagodeti *client server* rešenja i poslovnih programa. Sada će projektanti sistema moći da standardizuju rad, obezbeda visoke performanse i integrišu sve aplikacije koje sa u njihovom centrima koriste. Više im lako da sakupe sve podatke kojima korporacije raspolaže na sto mesto i tako znatno brže odgovore na zahteve onih kojima je analiza tih podataka najpotrebnija"

To je ono čemu se teži a prvi korisnici NT-a primetili su i druge stvari – pre svega velike hardverske apetite ovog paketa. Minimalna hardverska konfiguracija na zvuči posebno nemoguće ali je za komforan rad potrebno neših 16 megabajta RAM-a, a ni disk od 500-tinak megabajta nije na odmet. Što se procesora tiče stvari nisu toliko kritične, tako da će i na skromnijim 386 računarima NT obezbediti udoban rad ako ima dovoljno memorije. Sve ovo prevazilazi mogućnosti mnogih vlasnika personalnih računara, a i oni malobrojni srećnici koji imaju potrebnu konfiguraciju i kupe *Windows* NT ubrzo će primetiti da to i nije operativni sistem primeren jednom računaru – pravu snagu dostići će tek u nekoj relativno složenoj mreži mada nije jasno ima li Microsoft „pravo" da očekuje mnoge čije će radne stanice imati po 16 megabajta memorije. Moraće očito mnogo da se radi na „klijent-server" rešenjima.

Na tome će svakako raditi i nezavisne softverske firme koje je Microsoft obradovao izvornim kodom NT-a koji je stavljen na uvid javnosti (sem je već polazi renomiranim institucijama kao što su Brown Carnegie Mellon MIT, University of California Berkley University of Washington Xerox PARC i Stanford). O vitkav potez je u svetu personalnih računara bio gotovo nezamisliv takva „velikodušnost" praktikuje se samo kod basnoslovno skupih programa za velike sistema i *public domain* softvera. Microsoft je na ovaj način olakšao razvoj NT aplikacija i utro put ka njihovoj univerzalnosti. NT po prvi put nija samo operativni sistem za PC-je već i za DEC Alpha budućе PowerPC računara i mnoge druge modela. Pa ipak, sva ta fleksibilnost za sada je samo mrtvo slovo na papiru – komercijalne verzije NT a za druge platforme očekuju se tek tokom dolazeće godine.

Najveća konkurencija NT u svakako je IBM-ov OS/2 koji je u značajnoj prednosti jer se pojavio godinu dana pre NT-a u proleće 1992 godine. To znači da su mnoga „dečija bolesti" koje će mučiti NT za OS/2 stvar prošlosti – sredinom 1993 izašla je verzija 2.1 koja je ispravila većinu nedostataka otkrivenih kod verzije 2.0. Pre svega pojačana je podrška raznim grafičkim karticame i drugim uređajima – sada možeta de budete gotovo sigurni da će sa OS/2 savršeno prilagodi vašem hardveru. Pouzdanost sistema je

znatno unapređena a za „zlu na trebalo" su uvedene i alatke za operativnik diska. Multitasking je dalje nepredovao kompatibilnost sa DOS i *Windows* aplikacija je uvećana, prijavljivanje grešaka detaljno revidirano dokumentacija dopunjena. I uz sve to OS/2 sasvim pristojno i sedi na sistemima sa 8 megabajta memorija. To je za raziku od NT-a operativni sistem koji je pisan za personalne računara što mu može biti kako prednost tako i nedostatak. U zavisnosti od toga da li će NT uspeti da „zavoji" neke druge platforme kao što je PowerPC ili DEC Alpha ili će završiti svoju karijeru na PC kompatibilnim sistemima. Budućnost će pokazati koja je od ovih koncepcija bolja – za sada bismo je već personalac preporučili OS/2 2 1

PROGRAMIRANJE MIŠEM

Novi operativni sistemi znatno su promenili i pristup programiranju na PC-ju – mada su šanse da ćete ubuduće dikucati EDIT TETRIS C i početi da popunjavate ekran svojim programom. Tako sa pokazuje se, na mogu racionalno pisati programi koji traba da rade u nekom grafičkom okruženju kad što je *Windows* ili OS/2. Novu pristup Microsoft je sproveo u delo novom izuzetnom generacijom programskih jezika čijem imenu prethodi reč *Visual* – za sada su na raspolaganju *Visual Basic* i *Visual C*. Svaki od ovih paketa ne čini samo kompajler – zapravo kompajler je postao deo čije delovenje gotovo i da ne vidite. Sve vreme ste u integrisanoj okolini u kojoj „crtate" menije, maske vinjete i druge elemente korisničkog interfejsa neposredujete ih po ekranu određujete šta će se sa čim prekapati i tako dalje. Kada završite taj posao, integrisana okolina će umesto vas napisati program, istovijajući vam samo da dodate kod koji „opaluzuje" pojedine stavke menija i de formirate finalni EXE fajl. Ko je pra samo par godine mogao i pomisliti da nestupaju dani u kojima ćemo umesto komandama nekog jezika programirati pokretima miša!

Borland je u „grafičko programiranje" krenuo na što klasičnijim putem – uz sve snage se oslenjujući sa na principe objektnog programiranja tokom protekle godine izdao je nekoliko verzija svog C++-a (krajem godine izašao je *Borland C++ 4 0*) i obradovao ljubitelje paskala paketom *Borland Pascal 7 0 with Objects* i jedan i drugi paket obezbeđuju pisanje *Windows* programa i opremljeni su potrebnim alatka ma za dizajniranje atraktivnih menija ikona, maski i ostalih elementa – pa ipak „pravog" programiranja tu ima vše. Činjenica de se čak i najjednostavniji *Windows* program ne kreju sviše ne stotine kilobajta i zvomog koda čini de se mnogi „plaše" *Windows* programiranja – srećom stvari nisu baš tako strašne kao što ne prvi pogled izgledaju jer dobar deo tog posla obavi sama integrisana okolina po vašim nelozima. Pa ipak u umeće *Windows* programiranja treba uložiti dosta vremena i truda – nije malo programera koji se i dalje pitaju ispleti li sa tu posao

Statistike pokazuju da otprilika 50% svetska programerska populacija piše programa koji se izvršava ju linterno u okviru korporacija, kod nas je taj broj verovatno mnogo veći sasvim blizak brojki 100% A *Windows* programi ne tom tržištu baš i nisu mnogo traženi obezbedila programeru jednu laku 486 mašinu nije neki prevelikl trošak ali zamenki nekoliko desetina stolina ili čak hiljade AT-a ječim kompjuterima neophodnim se udoban rad pod *Windows*-om je jako teška odluka. Zato se za potrebe korporacija i dalje pišu DOS programi pa i Borland i Microsoft će i njihovi konkurenti za koje u našim krajevima rade čujemo recimo Zortech) čina sve što mogu da njihovi „gigantski" jezici obezbede i generisanje DOS koda. Ideal je zapravo integrisana okolina koja će radili pod *Windows*-om ili OS/2 obezbeđujući taka programeru sve zamisive alatke i pun komfor a na kraju po potrebi generisati sasvim običan EXE program koji radi kako u DOS prozoru tako i baz podrške *Windows*-a. Oba konkurenta to rade veoma dobro sli se ime ulaak da je Borland za korak ispredinjačio pre svega zahvaljući tome što njegovi proizvodi dobro sarađuju sa razmi DOS ekstenderima pa tako manje ili više lnesmo (ograničenje od 640 kilobajta sve se manje primećuje) koriste svu memoriju koju nadu na ciljnom računaru Borland osim toge svoje jezike prilagođava i OS/2 dok Microsoft verovatno iz nekih marketinških razloga, praktično ignoriše verziju 2 ovog operativnog sistema. Programerima ostaje samo da sa obrate IBM u ili (u poslednje vreme) Borlandu za svoj primerak C kompajlera

KAKO RASTU VERZIJE

Ne programiraju naravno svi na C-u paskalu bajaku III uopšte nekom višem programskom jeziku Sva su popularniji takozvani generaton aplikacija, od nosno paketi pomoću kojih se bez previše „tizičkog posla" može napisati program koji obavlje kompletan posao u nekoj oblasti pogotovu ako se ta oblasi od nosi na baze podataka uneksrsna izračunavanje po slovnu grafiku ili neki „konglomerat" tih primena Pored kod nas jako popularnog *Clipper*-a (firmu Nantucket kupio je Computer Associates International pa emo polovinom 1993 dobili i *Clipper 5.2*) sjajan posao radi i *FoxPro* ove godine dobili smo nove verzije za DOS i ze *Windows* koje uključuju izuzetno fleksibilno integrisano okruženje „automate" za formiranje i zvežaja i ekrana pravo dblje naredbi i funkcija izuzetne performanse i kvalitetnu *on line* i „papirnatu" dokumentaciju U „gru" se uključio i Microsoft paketom *Access* (pojavila su se verzije 1 0 i 1 1) Io je pravi dBASE nove generacije pisan specijalno za *Windows* Ne samo da u slogove možete uključivah crteže skeniranе fotografije i druge grafičke elemente već je i samo programiranje u jeziku nazvanom *Access Basic* jako slično *Windows* programiranju u *Visual* jezicima a ipak znatno pojednostavljeno i prilagođeno željenoj primeni

Microsoft Access je vredi i to reći jeden od jako ralkih programa koji se prodaju u verziji 1 0 II 1 1 – iz godine u godinu zapravo ima sve manje novih paketa Umesto toga iede se nove verzije poslojećih – malo je velikih programa koji nisu obeleženi „bar" kao 5 0 ili 6 0 a nade se i poneke „dvocifrena" verzije recimo *AutoCAD* 12 0 kome smo se lakoda ove godine obradovali Programa je očigledno toliko da je jako teško smisliti nešto što bi obezbedilo sasvim novu (e i pak opšte korisnu) primenu računara – softverskim firmama ostaje samo da usavršavaju postojeće proizvode e konsnicima da to usavršavanje prate koliko mogu A često ih na prate baš previše – na radi se u vek o novcu koji treba izdvojiti za novu verziju pa ni o tome što za njenu primenu treba kupiti novi tri puta veći disk nego o tome da sa čovek navikne na neki program i de mu je teško de se nevikave na njegovu „sledeću generaciju" kade i stara sasvim lepo radi posao Autor ovoge teksta na primer „po dužnosti" kompjuterskog publiciste „mora" de drži korak se novim verzijama softvera de ih isprobava uporabljuje prilagođeva pa ipak nešto od toga po napisanom prikazu „nestena" se moge diske u koriel prethodna verzija Teko koristim DOS 6 0 QEMM 7 2 *WordPerfect 6.0 Corel 4 0 Harvard Graphics 3* all i „antičke" *PC Tools 4 30 Venturu 3 0 Turbo Pascal 5 5, Clipper 5 1* Možde je kod drugih korsnika taj trend još izraženiji Kako programeki paketi rastu i postaju složeniji tako će i argumenti proizvođača morati da budu sve ubedljivij da bismo se mali novčanika!

CD-ROM I MREŽE

Trenutno su dva najubedljivija argumenta CD ROM-ovi i umrežavanje Mogio bi sa zapravo reći da je 1993 godine protekle u znaku CD ROM-a gotovo da nema firme koja nije pokušala da oformi neku multimedijalnu bazu podataka i distribuira je na ovoj na čin Velika konkurencija i činjenica da se CD ROM-ovi znatno lefe piratuju od diskata oborile je cene tako da su postale gotovo prihvatljive čak i za naš džep bar kade se radi o univerzalnim enciklopedijama i baza podataka. Podaci koje je teže neći i za koje je zainteresovan uži krug stručnjaka postižu mnogo više cene pa smo tako (na žalost samo u ogledima) videli CD ROM-ove koji koštaju desetak hiljada dolara Oni koji sebi na mogu da priušte ovakav CD ROM na trebaa da guba nadu da mnogih padataka tog tipa može se doći i na druge načine u komunikacji se svetom!

Komunikacija sa najčešće obavlja preke neke mreže – 1993 godine je afirmesla umrežavanje na svim nivoima Pojavio se pre svega, *Windows for Workgroups* koji obezbeduje nekolicini kolega u jednoj firmi da razmenjuju programe podatke, pisma, napomene i uopšte da posluju jedni sa drugima na mučući se od svoje tastature Za veće lokalna mreža zalao je *Novell Netware 4 0* koji je ponudio veću sigurnost podataka brži rad neku vrstu DBLSPACE-a (odnosno *on-line* kompresiju podataka na mrežnim diskovima) i što mnoge najviše interesuje bolju vezu sa svetom odnosno sa TCP/IP i drugim mrežama Jer kompjuterski svet je postao jedna ogromna mreža – globalni *Internet* je na primer 1991 godine počeo se 213 računara 1992 godine je u njege bilo uključeno 313 000 krajem 1991 godine 617 000 a u julu 1993 godine 1 776 000 kompjutera sa svih mendijana. No ih 1 776 000 računara sa možde 20 III 30 miliona korisnika nije konačna cifra – tu su i druge mreže keo što je moćni (ali i veoma konzervativno vođeni) *CompuServe* zatim IBX GVC akademske mreže kad što su *Bitnet* i EARN i mnogi mnogi drugi Svet zahvaljujući računarskim komunikacijama zaista postaje globalno selo svako je sa svakim u vezi svako svakome može da pošalje poruku da učestvuje u radu najrazličitijih stručnih konferencija da po ceni lokalnog telefonskog razgovora salima diskutuje se kolegom na drugom kontinentu

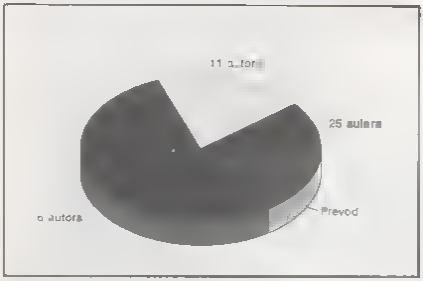
A KOD NAS...

Svako osim nas Zbog sankcije smo odsečeni od EARN-a sa kojim smo prethodnih godine komunicirali a nije nam omogućeno ni da ostvarimo svoj *Internet* domen Pa ipak nismo sasvim na rapu događaja prošle godine proradio je Beointernet, internet mreža VAX i PC računara ne FON-u i ETF-u sa kojom kao pridruženi član komunicira i naš Sezam Razmenjujemo poruke i konferencije po zemlji a, koliko sa može i sa inostranstvom Uvek se govori da Jugoslavija u svim tehnologijama „kaske" za svetom po dvadesetak i više godina Ispravnije bi bilo reći u svim tehnologijama osim kompjuterskih!

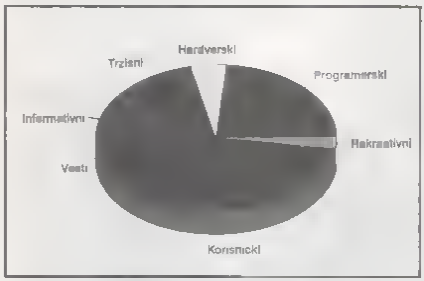
Domaće PC tržište i pored bede inflacije sankcije i svih drugih problema i dalje funkcioniše Jaka konkurencija čini de cena budu relativno niske sa avim na nivou svetskih bar kade se radi o međunarodnim pločama diskovima karticama i drugim standardnim komponentama Skupi su mahom kabasti elementi neročito monitori pa pak solidan broj vlasnika računara je tokom ove i prošle godina prešao na VGA kartice i monitor u boji ili bar monohrom VGA monitor Loša je jedino situacija se softverom – tržište koje je počelo da se formira 1990 i 1991 godine (mnogi su tada iskonstili pnliku da legalizuju svoje tekst-procesore baze podataka, kompajlere i druge programe koje svakodnevno koriste) sade zamira a softversko piratstvo ponovo cveta Činjenica da sa za male pare III besplatno dobijaju vrhunski strani programi ali često u nekompletnim verzijama i bez pratećа dokumentacije ujedno onemogućava formiranje tržišta za domaće programe velika većina programere piše baze podataka u *Clipper*-u po narudžbini dakle za jednog ili nekoliko korisnika iz meseca u mesec vodeći „ral" se nulama iza raznih iznosa koje sa pojavljuju pa nestaju Kvalitet tih programa je vrlo različit, ali imamo utisak da sa dodri programi i dalje mogu relativno pristojno prodati za programere i dalje nema zime keko ovde tako ni u daleksim zemljama u koje ih sve ovo što nam se dešava teko uspešno tera

RAČUNARI 1993. GODINE

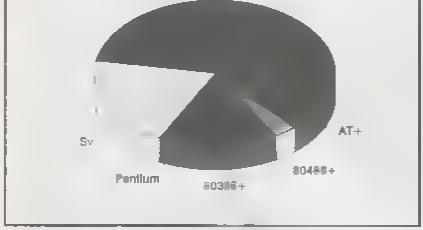
Dok je aktivnog kompjuterskog tržišta bilo veloa i za operativnik i napredak „Računara" koji su da sebi u ove rođendanske dane dopustimo malo neskromnosti jedan od značajnih činilaca koji su to tržište i formira Počeli smo od računara „Galaksije" koji je svaki naš čitalac mogao sam da napravi i koga je 15-tak hiljada naših čitalaca zaista i napravilo Bavili smo se osmobitnim računarima igrali se na njima, programirali hakerisali i osetili pravi trenutak kade je računarstvo trebalo ekretati prema profesionalnijim vodama Promovisali PC računara pokazali smo da kompjuter na mora da buda samo igračka i da na mora postojati samo da bi se programirao – računari su ušli u stotine domova i firmi oslakšavajući i ubrzavajući posao Svu su opremu trebalo je kupiti – organizovali smo nabavke iz Minhena i Singapura obezbedili renomiran forum za oglašavanje i zagarantovali oglašavača da će njihova ponuda doći u ruke profesionalaca koji o-



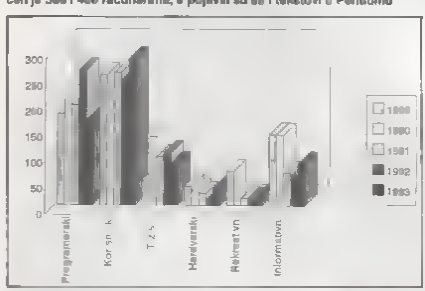
Slika 9 – Ko je ko u „Računarima" šest saradnika „nose" čak 54 posto časopisa. Ove godine bilo je čak 18% prevoda



Slika 7 – U korist korisnika računara koliko je tokom 1993. bilo korisničkih, programerskih, tržišnih i drugih tekstova



Slika 8 – Prema jačim konfiguracijama sva veći broj tekstova posvećen je 386 i 486 računarima, a pojavili su se i tekstovi o Pentiumu



Slika 6 – Šta interesuje čitaoce raspodela tekstova po temama tokom zadnjih 5 godina

dlučuju o nabavci. Pratili smo događaje na svetskom tržištu predstavljali najnovije verzije programskih paketa, prilagođavali ih domaćim potrebama, sortirali YU slova računali sa ogromnim i znosima čuvali se od virusa. Pre četiri godine uveli smo čitaoce u svet računarskih komunikacija otvarajući Sezam danas bez konkurencije najveći sistem te vrste u zemlji koji je po kapacitetima blizak najvećim evropskim BBS-ovima.

Tokom ove godine izaći će 9 brojeva „Računara" jedan manje nego što smo planirali. Pa ipak prosečan broj „Računara" (slika 1) bio je „debeli" nego ikada ranije – preko 116 strana. Oglasa smo i porasli sve ove krize imali koliko i prošle godine – 36 strana po broju (slike 2 i 4). Objavili smo 236 tekstova (slika 3) i 129 vesti iz pera 42 autora.

ŠTA KAŽU BROJKE

Najveći segment „Računara" nekih 285 strana posvećen je korisnicima (slika 7) – objavili smo uputstva za upotrebu DOS-a 6.0 i popularnih programa *Ventura Publisher*, *WordPerfect for Windows*, *WinWord*, *AutoCAD* i *CorelDraw*, prikazali razne verzije *Windows*-a i najnovije verzije programa *Ventura Publisher*, *WordPerfect AmiPro*, *Paradox for Windows*, *Access Wizdom*, *Adobe Photoshop*, *AutoCAD*, *Corel Draw*, *Designer*, *Harvard Graphics*, *WinFax Pro*, *Novell* QEMM, *PC Tools* za DOS i *Windows*. Nismo zaboravili ni domaće programe najviše klase kao što su *Tipotek*, *Novac*. U korisničke tekstove ubrajamo i one koji se bave komunikacijama – upoznali smo puteve kojima se, i pored sankcija, može stići do željenih stranih sistema i predstavili sve važne usluge svetskih mreža.

Dok je udeo korisničkih tekstova porastao u odnosu na prethodnu godinu (slika 6), programerskih tema je bilo nešto manje 185 strana ili 24%. Glavne programerske teme svakako su bili naši umeci „Microsoft C/C++ 7.0 struktura i funkcije" i „Moje škole paskala". Zapažani su i prikazi paketa *Visual C++ 1.0*, *Visual Basic 2.0*, *Borland C++ 3.1*, *Clipper 5.2* i *Blinker*, uvod u objektno programiranje u C++-u serija tekstova o programiranju mikroprocesora 80486 „prva pomoć" onima koji žele da njihovi programi komuniciraju sa mišem ili modemom, opis BTRIEVE-a i njegovih funkcija, kontraverzni *Paradox Engine*, uputstvo za pisanje rezidentnih programa. Nismo zaboravili ni „klasične" programerske teme – detaljno smo se bavili algoritmima za sortiranje i pretraživanje i upoznali mnoge tajne kalendara. Tu je naravno i naša redovna serija „Moje škola C-a", savetnici za *Unix* i *Clipper* (objavili smo i više opširnih tekstova o programiranju na *Clipper*-u i uz njih brojne biblioteke koje možete uključiti u svoje programe). Programerskom segmentu „Računara" nedostaje više tekstova o pisanju OS/2 i *Windows* programa – objavili smo samo jedan dosta relativno opširan.

Tekstova koji se bave tržištem bilo je otprilike koliko i prethodnih godina. Nešto smo reda prikazivali gotove sisteme ali smo zato detaljno proučavali komponente – „rat" mikroprocesora Pentium PowerPC i DEC Alpha, razne diskove koji mogu da se nađu na tržištu kao i interfejse za povezivanje sa njima, monitora, video kartice, generatora zvuka, brze modeme, štampače (naročito *Hewlett-Packard*-ovu porodicu laserskih štampača koji nose broj 4). Predstavili smo čak i nekoliko računara koji po mogućnostima (ali i ceni) znatno prevazilaze potrebe velike većine korisnika (npr. SUN SPARCstation, DECPC i Compaq SystamPro LT). U tekstove godine ubrajamo strog i opsežan brzinski test svih 486 računara koji se kod nas mogu nabaviti kao i umetak „Savršeni PC" iz „Računara 95" – oni koji planiraju nabavku opreme teško da će igde naći detaljniji i kvalitetniji vodič za izbor i konfigurisanje računarskog sistema.

Tekstove koji se bave hardverom bilo je gotovo tri puta više nego prethodne godine, ali ne taj način nije prevaziđen „rekord" od 5% hardverskih tekstova u „Računarima". Najzanimljivije teme odnosile su se na mikroprocesore i interfejse detaljno smo upoznali SCSI, SCSI 2 i *Local bus* i pomenuli karakteristike skoro svih magistrala kao što je EISA. Predstavili smo i „egzotičniji" hardver na primer fluptičke diskove i CD ROM-ove. „Ubičajena" 3% „Računara" posvećena su rekreativnom programiranju – uporedili smo popularne šahovske programe i objavili davet Pitalica različitih teških naročitu pažnju izazvala su rešenja problema iz oblasti kriptoanalize.

Preostalih 12% „Računara" popunili smo tekstovima koje, u nedostatku boljeg termina, nazvamo informativnim. To su pre svega „pratioci sadržaji" naših nagradnih igara „Combradova božićna jelka" poček kom godine a zatim velika nagradna igra „Sagni se i uzmi" čije rezultate objavljujemo u ovom broju „Računara" nejsrećniji čitaoci podelili su nagrade vredne preko 45 000 maraka. Tu su naravno i drugi sadržaji iz domaćeg i svetskog računarstva – bilo je istinamenje uslova da se posete inostrani sejmovi i slične manifestacije ali smo zahvaljujući elektronskoj pošti bili relativno dobro informisani o svemu što se dešavalo. Nismo se naravno ograničili samo na događaje – objavili smo zapažene intervjue sa Milenom Božićem Polom Kmetom, Vlastimirom Matejićem i Milanom Miljićem kao i umetak „Snažni poriv" koji se bavio životnim pričom Bila Gejtsa, osnivača i predsednika korporacije Microsoft.

Slika 8 pokazuje kako se u računarstvu (pa prema tome i u „Računarima") stvari brzo menjaju – prva „torta" opisuje kompjutere o kojima smo pisali u naših prvih 10 brojeva a druge je posvećena proteklo] godini. Nekada su vidimo večnu „torte" suverano držali *Spectrum Commodore Galaksije* i BBC dok je PC-ju posvećeno jedva pola procenta „Računara". No to parčence je za ovih 10 godina prerasio u čitavu tematsku „tortu", koju sada delimo na komade zvane AT, 386, 486 i počavši od ove godine Pentium.

POGLED U BUDUĆNOST

Hoće li tokom sledećih godine parčence zvano Pentium prerasti u čitavu tortu ili će možda *PC* računari „izumreti" i ustupiti svoja mesta PowerPC-ju, DEC Alphi ili nekom kompjuteru koji još nije ni napravljen kao što su *Spectrum 48*, *Commodore 64* i *Amstrad 6128* ustupili mesta njima? Danas ne to pitanje teško da iko može odgovoriti. Znamo samo jedno „Računari" će vas i tokom sledeće decenije obaveštavati o svim tim događajima trudeći se da vam pomognu da što racionalnije konfigurišete i primenite svoje kompjutera. Kada smo već na početku ovog poglavlja zaboravili ne skromnost recimo da ono što smo postigli tokom ovih deset godina i veliko poverenje koje nam iz meseca u mesec ukazuju čitaoci budi nadu da ćemo u sledeći vek ući kao jedan od značajnijih evropskih kompjuterskih časopisa. Naša mala Redakcije i ekipe spoljnih saradnika učiniće sve da „Računari" takvo mesto i zaustimo ako nam stanje u koma će se ova zamlja nalazil pruži barem najmanje uslove za to.

1 **Aleksić Vladan**
1 Turbo na dek noga (G1800 ploter) (93/18)
2.Veća brzina uz manje napora (ACAD 12) (93/11)

2. **Bašić Milan**
1 Novi OS/2 menadžer videti (92/26)
2.Visual Basic 2.0 (93/24)

3 **Batoćanin Nenad**
1 Clipper savetnik (93/71)
2.Clipper savetnik (93/71)
3 Clipper u mreži (94/56)
4 Formatiranje teksta (93/80)
5.Kao oka trepalj (Blinker) (94/32)
6.Knjige u kojoj sve piše (93/82)
7 Moja škola Clippera (93/62)
8.Novi Clipper (5.2) (93/35)
9.Objekti za nestrpljive (93/69)
10 Povratak keramike (xClip 3.0) (93/46)
11 Promenljive u Clipper u (93/63)
12.Vreme je (za) NOVAC (94/22)
13 Začinti po ukusu (Multi-Edit) (91/42)

4 **Brkić Zoran**
1 Duel (ChessMaster vs. Stalker) (93/62)
2.Šahovski pomoćnik (Chess Assistent) (93/37)

5. **Bulajić Jovan**
1 Do pretprošlog boljta (na disketi) (94/43)
2.SezamFile (93/78)
3.SezamFile (91/81)
4 SezamFile (93/80)
5.SezamFile (93/81)
6.SezamFile (94/66)
7 SezamFile (94/60)
8 SezamFile (93/73)

6 **Čurin Ante**
1 Najbolji prijatelj prijatelj (AMI PRO 3.0) (93/40)

7 **Ćosić Veera**
1 Ala – kvartni sleak (Pet Kmeta) (93/36)
2.Moj virtuelni svet (Milan Bošić) (93/66)
3.Nauka ne stanuje ovde (V. Matejić) (93/20)
4 Nikad više sam (Milan Mijić) (93/52)
5.OSA u Novom Sadu, Sibiru i na Kipru (93/23)
6 Ujrkoš svima, uprkos svemu (Sajam tehnike) (93/17)

8 **Dinčić Dejan**
1 Kako da uradim Windows aplikaciju (93/72)
2 Poslovne knjige (94/58)

9 **Dumeljić Dragan**
1 Obračun keš O.K. lokala (93/15)

10 **Jelović Dejan**
1 Dejan nije kod kuće. (93/31)
2.Grafici objekti (94/64)
3 Komarci ili'e neba (traženje grešaka) (91/70)
4 O stringovima i ljudima (93/60)
5 Objekti u praksi (94/70)
6 Prelaz u prednjičijem (93/95)
7 Prodavci zdravog pamet (consulting) (93/53)
8 Put do uspeha (94/62)
9 Sudar svetova (93/95)
10.U kom gmiu leži beg? (94/56)

11 **Jevanović / Klisić**
1 Slogovi i polja (Paradox Engine) (93/56)
2.Zamenite meter ako su podaci sporl (93/72)

12 **Jovanović Milijan**
1 Ako vidite mišave (93/56)
2 Kontrola sistema (93/66)

13 **Jovanović Miroslav**
1 CorelDraw (93/ 0)
2 Nov kao Corel (4.0) (94/48)

14 **Jovanović Pavica**
1 Tehničica za zemljanke (Fonts 3BSE) (92/90)
2 Ako je lepo, mogu je kažu (Olimpijada) (93/20)
3.Hepening u Hepiu (93/ 0)
4 Miševi za najbolje YU programere (93/24)

15 **Kehler Zoran**
1 Mreže za mase (Win for Workgroups) (91 51)
2.Stvaranje PC (93/ 0)
3.Usluge za usluge (Win for Workgroups) (93 40)
4 WIN.INI (94/77)
5 WIN.INI (93/70)

16 **Kostić Vlada**
1 Da li je neko video Tanju (pretraga teksta) (93/64)

17 **Marinković Aleksandar**
1 AutoCAD (91/ 0)

18 **Matić Spiridon**
1 Heterogena mreža (94/31)
2.Server po levi (Compaq SystemPro LT) (94/23)

19 **Mihajlović Bojan**
1 Kvarljivi šlepovi (93/77)

20 **Mijanović Srđan**
1 Lego jezik (Maneuvre) (93/87)

21 **Nejgebauer Ivan**
1 Zbogom notepad (MicroEMACS) (91/46)

22 **Novaković Nebojša**
1 Lep, snažan, kompaktan (SUN SPARCstation) (93/11)

23 **Pelević Pavle**
1 Ipak za kraće (93/19)
2.Moja škola Unix-a (93/83)
3.Moja škola Unix-a (93/74)
4 Mudrac iz tarzala (Watcom) (93/42)

24 **Petrović Bojan**
1 Balansirana stabla (94/66)
2.Elementarne metode (93/56)
3.Elementarne metode (pretraživanja) (93/76)
4 FileFind za početnike (91/72)
5 Hufmanove stabla (94/60)
6.Jedan pravi TSR (93/64)
7 Ko umije njemu sviju (kompresija) (95/52)
8 Laki uvod u DPMI (93/58)
9 Memorija i modeli (C) (93/57)
10.Napredne metode (pretraživanja) (91/83)
11 Sortiranje u ploči (93/80)
12.Sve za vrt oko oka (93/58)
13.TSR uzvraća udarac (93/80)

25 **Popović Dušan**
1 Bolje od oka (Paradox for Windows) (93/40)
2.Bez prave konkurencije (Borland C++ 3.1) (94/36)
3 Iz paradoksa u paradoks (93/26)
4 Jaša za po kući (Visual Basic 2.0) (93/45)
5.Ko programira taj i gradi (Turbo Debuger) (91/40)
6.Microsoft Word 2.0 (93/ 0)
7 Rastegljivi dizajn (Stacker) (93/62)

29 **Regasek Jovan**
1 Jaka četvorka (Lasarjet IV) (93/14)
2 Na pola puta (Cx486DLC) (94/ 0)
3.Sad ili nikad (AEG LP 90) (94/12)
4 Sex slici slaje uzrujanje (Superskech 3D) (93/13)
5.Super flapi (floppiji dizajn) (94/21)
6.Temperamentni štedlja (HP LJ 4L) (94/10)
7 U najlepšoj kategoriji (test 80486/50) (94/14)
8 Zvezda u uzletu (Star ZA-250) (94/28)

27 **Ristanović Dejan**
1 Adresni modovi (93/74)
2.Alfa – sleak ali kakav? (93/16)
3 Argh! (94/66)
4 Bajtovi lične prirode (93/64)
5 Bajtovi lične prirode (93/78)
6 Bajtovi lične prirode (91/78)
7 Bajtovi lične prirode (94/78)
8 Bajtovi lične prirode (93/74)
9 Bajtovi lične prirode (93/87)
10.Bajtovi lične prirode (93/85)
11 Bajtovi lične prirode (93/76)
12.Ceo svet je vaš (Internet) (93/41)
13.Da li će jasti Pentium (PowerPC)? (93/12)
14 dBASE devedesetih (MS Access) (93/38)
15 Devet tom („Računari" 1992. godine) (94/91)
16.Dragnicke svej DOS svet (DOS 6.0) (91/32)
17 Dok su drugi spavali (ARJ 2.41) (94/36)
18 Indeks tema (93/95)
19 Interapt mehanizam 80486 (94/53)
20.Istorijska Pisalica (93/75)
21 Moje škole početak (94/ 0)
22.MS DOS 6.0 (94/ 0)
23 PKZIP 2.04G (94/ 0)
24 Peške pod sankcijama (94/54)
25 Povratak u DOS (WF 6.0) (93/38)
26.Preko hleba QEMM (7.0) (93/28)
27 Problem slugih faktorijela (93/74)
28.Razgrani bitovi (93/72)
29 RISC Pisalica (93/72)
30 Set instrukcija 80486 (1) (94/62)
31 Set instrukcija 80486 (2) (91/56)
32 Sezam Biten (93/66)
33 Sezam Biten (93/68)
34 Sezam Biten (94/50)
35 Sezam Biten (91/76)
36 Sezam Biten (92/76)
37 Sezam Biten (93/76)
38 Sezam Biten (93/64)
39 Sezam Biten (93/78)
40.Sezam Biten (93/80)
41 Strega po kalendaru (94/78)
42.Šahovska Pisalica (93/84)
43 Šta smo lupali kad smo bili mali (93/52)
44 Šestovi godine (93/78)
45 U malom džepu (Palm Organizer) (94/37)
46 U modi su dlinovi (Borland Pascal 7.0) (93/31)
47 U oznoj rundi (PC Tools 8.0) (91/34)
48 U srcu broja 7 (93/60)
49 Ventura (4.1) u boji (94/42)
50 Višeprogramski rad 80486 (94/58)
51 Windows Pisalica (91/85)
52.WordPerfect for Windows (93/ 0)
53 Zaštita memorija (94/53)
54.ZIP (2.0) je stigao (93/52)
55 Zvezde je rođene (Pentium) (93/26)
56 Žit Venov problem (93/60)

28 **Stakić Miodrag**
1 Fax ko od šale (WinFaxPro 3.0) (94/44)

29 **Stamenović Vladimir**
1 Quattro... pa svi ostali (93/38)

30 **Stanišić Darko**
1.Alhemija ili stvarnost (konverzija graf. formata) (93/44)

31 **Šušnjar Aleksandar**
1 PC onomatopeja (generisanje zvuka) (93/52)

32 **Tepavčević Bojan**
1 Koliko brzo računaju (test sa Sajma) (94/22)

33 **Tešić Dejan**
1 Jedanaest prestupa u pet koraka (93/66)

34 **Titov Jurij**
1 Drži'e sliku (Hijack) (93/31)
2 Elektronski Weber (94/26)
3.Fentregrapher (94/40)
4 Putovanje kroz ljudsko telo (94/40)

35. **Tomić Željko**
1 Tilina, snima se (93/48)

36. **Velimirović Milan**
1 Kralj lasera (LaserMaster 800) (91/22)
2 S druge strane ogledala (PS/TrueType) (91/96)

37 **Veselinović Dejan**
1 Čelni jahača (INFO-D) (94/25)
2.I slaje se svetom (sleheri) (91/26)
3 Inteligentni kontroleri (93/66)
4.Leksi je stigao (93/26)
5.Može li manje (portabl računari)? (91/26)
6 Ubrzavanje diskova (93/47)
7 Vaše prakne (CeBit) (94/10)

38 **Vlajić Saša**
1 Prvi ljubav drugi put (Plustort) (93/80)

39 **Vukelić / Blagojac**
1 Priča o slovu 2 (Ventura) (93/46)
2 Ventura Publisher (93/ 0)

40. **Vukuvić Dragutin**
1 Mreže na nov način (NetWare 4.0) (94/14)

41 **Životić Zoran**
1 Berba za modeme (94/74)
2.Čarolnjak iz Oza (Visual C++ 1.0) (94/36)
3 Čista veza sa svetom (modemi) (93/22)
4 Lekcija I ETRIEVE funkcije (91/48)
5 Lekcija II ETRIEVE u praksi (92/60)
6 Memorija na klo (94/76)
7 Moja škola C-a (93/95)
8 Moja škola C-a (93/80)
9 Moja škola C-a (93/73)
10 Moja škola C-a (91/75)
11 Moja škola C-a (93/73)
12 Moja škola C-a (93/60)
13 Moja škola C-a (94/91)
14.Moja škola C-a (94/73)
15.Moja škola C-a (94/76)
16 MS C/C++ 7.0: strukture i funkcije (94/ 0)
17 Sistemsko sanduče (environment) (94/70)
18 Tipografija kroz igru (Tipetak 2.5i) (93/51)

1 **Adaptacija komercijalnog softvera**
1.Priča o slovu Ž (Ventura) (00/ 48)

2. **Alatke**
1 Centralni bolični jelka (00/ 6)
2 Hepening u Hajatu (00/ 6)
3 Računar CT34S5X skriven iD Studiš iz Zamuna (00/ 51)
4 Svi naši spenzori (93/ 83)

3 **Algoritmi**
1 Satenarana stabla (00/ 58)
2 Elementarne metode (93/ 56)
3 Elementarne metode (pretraživanje) (00/ 78)
4.Hufmanovo stablo (00/ 60)
5.Ko umije njamu dvije (kompresija) (00/ 52)
6 Najnovije metode (pretraživanje) (01/ 63)
7 Sortiranje u stvoji (00/ 50)
8 Stroge po kalendaru (94/ 76)

4 **Asemblersko programiranje**
1 Adresni modovi (00/ 74)
2.Interapt mehanizam 80486 (00/ 53)
3.Set instrukcija 80486 (1) (00/ 62)
4.Set instrukcija 80486 (2) (01/ 35)
5 Višeprogramski rad 80486 (00/ 56)
6.Zaštita memorije (94/ 53)

5. **Baze podataka**
1.Baze od oka (Paradox for Windows) (00/ 46)
2 Clipper savetnik (00/ 71)
3 Clipper savetnik (00/ 71)
4 Clipper u mrtvi (00/ 65)
5.dBASE developeasth (MS Access) (93/ 38)
6 Formatiranje teksta (92/ 60)
7 Is paradoksa u paradoksa (00/ 28)
8.Kao oka trepta) (Blinker) (00/ 30)
9 Knjiga u kojoj ave piše (00/ 32)
10 Lekcija I IBTRIEVE funkcije (01/ 48)
11 Lekcija II: BTRIEVE u praksi (00/ 66)
12 Moja škola Clippera (00/ 82)
13 Mudrac iz Izraela (Wisdom) (93/ 42)
14 Novi Clipper (5.2) (00/ 36)
15 Objekti za nestrpljive (00/ 46)
16 Pavratak korenima (dClip 3.0) (00/ 46)
17 Premenljive u Clipper-u (00/ 63)
18 Singovi i polja (Paradox Engine) (00/ 56)
19 Šahovski pomoćnik (Chess Assistant) (00/ 37)
20.Zamenite motor ali ni podesi spori (00/ 78)

6. **Budućnost računara**
1 Borba za vrhu linju (softverski trendovi) (00/ 11)
2 Put do uspeha (00/ 62)

7 **Funkcionisanje računara i opreme**
1 Bas na stanuje ovde (Local Bus) (00/ 22)
2 Lekcija I: sligao (00/ 86)

8. **Grafika i animacija**
1.Alhemija 6 stvamost (konv graf formata) (00/ 44)
2.AutoCAD (91/ 6)
3 CorelDraw (93/ 0)
4 Designer 4.0 bolji od Canela (00/ 24)
5 Harvard Graphics Windows 2.0 (00/ 24)
6.Kako da uradim (Windows aplikaciju) (00/ 72)
7 Niko kao Corel (4.0) (00/ 46)
8 Photoshop otvara prozore (Adobe Photoshop) (00/ 45)
9.Pravi izbor u svi promena (video kartice) (01/ 10)
10 Sta sliči deja ubrzanje (Superstation 3D) (00/ 13)
11 Silikona akrobatika (Adobe Acrobat) (00/ 30)
12 Tilina, anima se (00/ 48)
13 Veća brzina uz manje napora (ACAD 12) (00/ 11)

9 **Igre**
1 Duel (ChessMaster vs Stalker) (00/ 62)

10. **Inostrani kompjuterski život**
1 Borba za vrhu linju (softverski trendovi) (00/ 11)
2.Cao svet je vali (Internet) (00/ 46)
3 Kako te rade Slovenci (92/ 11)
4 Nova DLE generacija (01/ 10)
5 Prodavci zdrave pameti (consulting) (00/ 53)

11 **Interfejsi**
1 Leteći diskovi (SCSI-2) (00/ 46)

12. **Intervjui**
1.Alfa – kvantni skok (Pal Kmeta) (00/ 36)
2 Moj virtuelni svet (Milan Bošić) (00/ 36)
3 Nauka ne stanuje ovde (V. Matejić) (93/ 20)
4.Nikad više sam (Milan Mijić) (00/ 52)

13 **Istorija računarstva**
1.Snažni poriv (Bill Gates) (00/ 0)
2 Šta ame lupati kad ame bili mali (00/ 52)

14. **Komercijalni softver**
1.Alhemija 6 stvamost (konverzija graf formata) (00/ 44)
2.Baza od oka (Paradox for Windows) (00/ 40)
3.Bez prave konkurencije (Borland C++ 3.1) (00/ 36)
4 Bila kuća Windows sveta (00/ 18)
5.Čarobnjak iz Oza (Visual C++ 1.0) (00/ 36)
6.Da li je neka video Tanju (pretraga teksta) (00/ 64)
7 dBASE developeasth (MS Access) (93/ 38)
8.Designer 4 0, bolji od Canela (00/ 24)
9.Degradira svej DOS svet (DOS 6.0) (01/ 32)
10.Dr2ta stku (Hijeli) (00/ 31)
11 Elektronski Webster (00/ 28)
12.Fax ko od šale (WinFaxPro 5.0) (00/ 44)
13.Fontmographer (00/ 40)
14 Geopoder desktopa (PC Tools for Windows) (93/ 50)
15.Harvard Graphics Windows 2.0 (00/ 24)
16.Iz paradoksa u paradoks (00/ 28)
17 Jaslk za pe kući (Visual Basic 2.0) (93/ 46)
18.Kao oka trepta) (Blinker) (00/ 32)
19.Ko programira taj i grali (Turbo Debugger) (01/ 40)
20.Kuda idu operativni sistemi (00/ 28)
21 Mudrac iz Izraela (Wisdom) (93/ 42)
22.Najbolji plišlev prijatel (AMI PRO 3.0) (00/ 40)
23.Napravi mi mala mesta (Stasker ltd) (93/ 31)
24.Niko kao Corel (4.0) (00/ 46)
25.Nova era tehnia (00/ 10)
26.Nove DLE generacija (01/ 10)
27 Novi Clipper (5.2) (00/ 36)
28 Photoshop otvara prozore (Adobe Photoshop) (93/ 46)
29.Pavratak korenima (dClip 3.0) (00/ 46)
30.Pavratak u DOS (WP 6.0) (00/ 36)
31 Preko hleba QEMM (7.0) (00/ 28)
32.Prenaditi pravu reč (Grammatik 6) (00/ 45)
33.Prva ljubav drugi put (Plusfon) (00/ 60)
34 Putevanje kroz ljudske lelo (94/ 40)
35 Quattro ...pa avi ostali (00/ 30)
36.Rasteglivi diskovi (Stacker) (00/ 60)
37 Rizmica znanja (Microsoft Encarta) (00/ 26)
38.Silikona akrobatika (Adobe Acrobat) (00/ 30)
39 Smena kraljevska straža (DTP program) (00/ 19)
40 Šahovski pomoćnik (Chess Assistant) (00/ 37)
41 Tipografija kroz igru (Tipotek 2.01) (93/ 51)
42.U modi su užnovi (Borland Pascal 7 0) (00/ 31)
43 U samoj rundi (PC Tools 8 0) (91/ 34)
44 Veća brzina uz manje napora (ACAD 12) (00/ 11)
45.Ventura (4 1) u beji (00/ 42)
46.Visual Basic 2.0 (93/ 24)
47 Vreme je (za) NOVAC (00/ 22)
48.Začinili po ukusu (Multi-Edit) (01/ 42)
49.Zbogom notepad (MicroEMACS) (01/ 40)

15 **Komunikacije**
1 Bezbe za modemima (00/ 74)
2.Cao svet je vali (Internet) (00/ 46)
3.Čista veza za svetom (modemi) (00/ 22)
4 Fax ko od šale (WinFaxPro 3.0) (00/ 44)
5.Felix pod senkoljama (00/ 54)
6.Seam Bihen (00/ 46)
7 Seam Bihen (00/ 80)
8 Seam Bihen (00/ 90)
9 Seam Bihen (01/ 78)
10 Seam Bihen (92/ 76)
11 Seam Bihen (93/ 78)
12.Seam Bihen (00/ 94)
13 Seam Bihen (00/ 76)
14 Seam Bihen (00/ 80)
15 SeamFile (00/ 72)
16.SeamFile (01/ 01)
17 SeamFile (00/ 80)
18.SeamFile (93/ 81)
19.SeamFile (00/ 82)
20 SeamFile (00/ 82)
21 SeamFile (00/ 73)

16 **Masovna memorija**
1 I dalje se svetom (diskovi) (01/ 38)
2.Leteći diskovi (SCSI-2) (00/ 46)
3.Rizmica znanja (Microsoft Encarta) (00/ 26)
4 SCSI je bio beli (00/ 18)
5.Super flopi (fleksibilni diskovi) (00/ 21)
6.Ubrzavanje diskova (00/ 47)

17 **Mikroprocesori i čipovi**
1 Adresni modovi (00/ 74)
2.Alfa – kvantni skok (Pal Kmeta) (00/ 36)
3.Alfa – skok ali kakav? (00/ 16)
4 Bas na stanuje ovde (Local Bus) (00/ 22)
5 Da li će pasti Pentium (PowerPC)? (00/ 12)
6 Intel 486 i svi njegovi klonovi (00/ 57)
7 Interapt mehanizam 80486 (00/ 53)
8.Kvarljivi čipovi (00/ 77)
9.Lokal je stigao (00/ 58)
10.Ne peta puta (Cx486DLC) (00/ 8)
11 Pentium iz tamelja menja PC svet (94/ 14)
12.Pevaćete pravi pesac (mikroprocesori) (00/ 66)
13.Set instrukcija 80486 (1) (00/ 62)
14 Set instrukcija 80486 (2) (01/ 35)
15.Višeprogramski rad 80486 (94/ 56)
16.Zaštita memorije (93/ 53)
17 Zvezda je rođena (Pentium) (93/ 28)

18 **Mreže računara**
1 Heterogena mreža (00/ 31)
2.Lekcija I ISTRIEVE funkcije (01/ 48)
3.Lekcija II: ISTRIEVE u praksi (00/ 66)
4 Mreže na nov način (NetWare 4 0) (00/ 14)
5 Mreže za masa (Windows for Workgroups) (01/ 51)
6.Server je krvi (Compaq SystemPro LT) (00/ 22)
7 Usluge za usluge (Win for Workgroups) (00/ 46)

19 **Nabavka kompjutera i opreme**
1 Savršeni PC (00/ 0)

20 **Obrada teksta na računaru**
1 Autoteperski vodič kroz veleprske fantove (00/ 42)
2 Elektronski Webster (00/ 28)
3 Fontmographer (00/ 40)
4 Microsoft Word 2.0 (00/ 0)
5.Najbolji plišlev prijatel (AMI PRO 3.0) (00/ 40)
6.Pavratak u DOS (WP 6.0) (00/ 36)
7 Priča o slovu Ž (Ventura) (00/ 48)
8.Prenaditi pravu reč (Grammatik 6) (00/ 45)
9.S druge strane ogledala (PostScript/TrueType) (01/ 58)
10.Smena kraljevska straža (DTP program) (00/ 19)
11 Tipografija kroz igru (Tipotek 2.01) (93/ 51)
12.TrueType od A do Š (00/ 46)
13.Ventura (4 1) u beji (00/ 42)
14.Ventura Publisher (00/ 0)
15.WordPerfect for Windows (00/ 0)
16.Začinili po ukusu (Multi-Edit) (01/ 42)
17 Zbogom, notepad (MicroEMACS) (01/ 40)

21 **Odjeci**
1 Jedanaest prestupa u pet koraka (00/ 58)
2.Obračun kod O.K. lokala (00/ 18)
3 Prestup s prednjim (00/ 00)

22. **Operativni sistemi**
1 Bila kuća Windows sveta (00/ 18)
2.Degradira svej DOS svet (DOS 6.0) (01/ 32)
3.Kuda idu operativni sistemi (00/ 28)
4 Moja škola Unix-a (94/ 30)
5 Moja škola Unix-a (00/ 74)
6.Mreže na nov način (NetWare 4.0) (00/ 14)
7 Mreže za masa (Windows for Workgroups) (01/ 51)
8 MS DOS 6 0 (00/ 0)
9 Neka ustane bolji (Windows – OS/2) (00/ 4)
10 Novi OS/2 menaće videti (92/ 36)
11 Sve se vrti oko dela (00/ 58)
12.Šestica (DOS) izrutra (00/ 68)
13 Usluge za usluge (Win for Workgroups) (00/ 46)
14 WIN INI (00/ 77)
15.WIN.INI (00/ 78)

23 **Performanse računara i softvera**
1 Kolike brzo računaju (test se Sajmić) (93/ 22)
2.U najlažej kategoriji (test 80486/50) (93/ 14)

24 **Periferijska oprema**
1 Čista veza sa svetom (modemi) (00/ 22)
2.Ekspicalja boja (monitori) (92/ 10)
3.I dalje sa svetom (diskovi) (01/ 38)
4.Jasno kao na danu (monitori) (00/ 16)
5.PC onomatopeja (generisanje zvuka) (00/ 52)
6.Pravi bolor u svi promena (video kartice) (01/ 10)
7 SCSI je bio beli (00/ 18)
8.Sile oka deja ubrzanje (Superstation 3D) (00/ 13)
9 Turbo na dve noge (G1000 plotar) (00/ 18)

25 **Pisalice**
1 Arghi (00/ 66)
2.Jednajska Pisalica (00/ 75)
3.Problem dugih fabrikanata (00/ 74)
4.Razigrani bitovi (00/ 72)
5.RISC Pisalica (93/ 72)
6.Šahovska Pisalica (94/ 94)
7 U znaku broja 7 (00/ 62)
8.Windows Pisalica (01/ 68)
9 Ži Vemov problem (00/ 66)

26 **Profesionalni hardver**
1 Obračun kod O.K. lokala (00/ 18)

27 **Programski jezici**
1 Bez prave konkurencije (Borland C++ 3 1) (00/ 36)

2.Čarolanjak iz Oza (Visual C++ 1.0) (96/39)
3.Dejan nije kod kuće (93/61)
4.Grafični objekti (92/64)
5.Jedanaest prestupa u pet koraka (89/98)
6.Jastuk za po kući (Visual Basic 2.0) (93/46)
7 Ko programira taj i greši (Turbo Debuger) (91/40)
8.Kemerci K'o nula (traženje grešaka) (91/70)
9.Lego jastuk (Manevra) (96/87)
10 Memorija i modeli (C) (93/57)
11 Moja škola C-a (96/85)
12.Moja škola C-a (89/88)
13 Moja škola C-a (94/73)
14 Moja škola C-a (91/76)
15 Moja škola C-a (96/73)
16 Moja škola C-a (93/85)
17 Moja škola C-a (96/61)
18 Moja škola C-a (96/73)
19 Moja škola C-a (96/76)
20 Moja škola paskala (96/ 0)
21 MS C/C++ 7.0: strukture i funkcije (94/ 0)
22.O stringovima i ljuskama (96/80)
23.Objekti u prozoru (94/70)
24.Prestup s prekomilijarom (89/98)
25.Prva ljubav drugi put (Plunkart) (96/80)
26.Sudar svetova (94/90)
27 U kom grmu leži bag? (88/58)
28.U medi su dlanovi (Borland Pascal 7.0) (96/31)
29 Visual Basic 2.0 (93/34)

28 Proširenje kompjutera
1 Inteligentni kontroleri (94/88)

29 Računari u izlogu
1 Beležnice za biznismene (Acros 386SE) (89/56)
2.Četiri jahača (INFO-D) (96/25)
3 Koliko hran računaju (test sa Sajma) (89/22)
4 Lep snažan kompjuter (SUN SPARCstation) (96/11)
5 Može li manje (prenosni računari) (91/26)
6 Personalni DEC (DECPC AXP 150) (96/17)
7 Server po krvi (Compaq SystemPro LT) (96/22)
8.U malom džepu (Psion Organiser) (92/37)
9.U najnižoj kategoriji (test 80486/50) (93/14)

30 Razglednice i reportaže
1 pak za kraće (93/18)
2.Uprkos svima, uprkos svemu (Sajam tehnike) (93/17)
3.Valar prolinu (Celik) (89/10)

31 Savetnik
1 Bajtovi lične prirode (89/84)
2.Bajtovi lične prirode (88/76)
3 Bajtovi lične prirode (91/76)
4 Bajtovi lične prirode (92/76)
5 Bajtovi lične prirode (93/74)
6 Bajtovi lične prirode (96/87)
7 Bajtovi lične prirode (96/85)
8 Bajtovi lične prirode (94/76)
9.Clipper savetnik (94/71)
10.Clipper savetnik (90/71)
11 Moja škola C-a (96/85)
12.Moja škola C-a (89/83)
13.Moja škola C-a (94/73)
14 Moja škola C-a (91/76)
15.Moja škola C-a (92/73)
16.Moja škola C-a (94/88)
17 Moja škola C-a (89/91)
18.Moja škola C-a (94/73)
19 Moja škola C-a (96/75)
20 Moja škola Clippera (94/92)
21 Moja škola Unix-a (94/93)
22.Moja škola Unix-a (88/74)
23 Promenljive u Clipper-u (96/83)
24 Šesovi godine (88/76)
25 WINJNI (94/77)
26.WINJNI (96/76)

32 Sistemska programiranje
1 Jedan pravi TSR (90/84)
2.Kontrola tastature (94/86)
3 Laki uvod u DPMI (96/56)
4 Memorija na kliz (96/70)
5 OLE bi A da Š (92/51)
6.Sistemsko okruženje (environment) (90/70)
7 Sve se vrti oko diska (96/58)
8 Šestica (DOS) struktura (89/56)
9 TSR uzvrata udarac (96/80)

33 Specijalne primene računara
1 Da li je neko video Tanju (prelepa lokota) (96/84)
2.PC anomaloskopija (generisanje zvuka) (96/52)
3.Putevanje kroz ljudsko telo (96/40)

Top lista autora svih vremena

Autor	Takstova
Dejan Ristanović	536
Zoran Životić	154
Dejan Veselinović	112
Žarko Berberski	95
Nevenka Spalević	74
Jovan Skuljan	67
Duško Savić	53
Marko Krić	45
Nenad Batočanin	43
Vlada Kostić	28
Vesna Ćosić	31
Žarko Vukosavljević	30
Dušan Slavić	28
Branko Đaković	27
Jovan Regasek	26
Nebojša Novaković	26
Vladimir Stamenović	22
Jelena Rupnik	22
Zoran Obradović	21
Dragutin Vuković	21
Dada Vujasinović	21
Jovan Bulajić	21
Bojan Petrović	20
Cerovski Viktor	20
Slobodan Perović	18
Voja Gašić	18
Dejan Dinčić	18
Aleksandar Radovanović	18
Blažimir Mišo	18
Davor Petrić	17
Radomir Mihajlović	17
Dejan Jelović	16
Anđelko Zgoralac	16
Pavle Peković	14
Saša Svitlica	14
Esad Jakupović	14
Vlada Stojiljković	13
Božidar Travica	13
Zoran Cvjetić	12
Željko Jurić	12
Branko Marović	12
Dušan Popović	11
Zoran Kehler	11
Miroslav Olanjin	11
Miodrag Polkonjak	10

Lista autora više od deset objavljenih priloga u periodu 1984-1993.

4 Televizija na računaru (91/15)
6 Tilina, anima se (96/45)

34 Štampači
1 Jaka četvorka (LaserJet IV) (96/14)
2.Kraj lasera (LaserMaster 800) (91/22)
3.Sad i nikad (AEG LP 80) (96/12)
4 Temperamentni štorlja (HP LJ 4L) (96/10)
5.Zvezda u oblu (Star ZA-250) (94/29)

35 Tabelarni proračuni
1 Nova era tabela (96/10)
2.Quattro pa svi ostali (88/38)

36 Tehnika programiranja
1 Ako volite mešave (89/62)
2.Balansirana stabla (96/85)

SPECIJALNI DODACI

MS C/C++ 7.0: strukture i funkcije	računari 88
Snažan perfv: Bil Gejts	računari 89
Microsoft Word 2.0	računari 89
PKZIP 2.04C	računari 89
WordPerfect for Windows	računari 90
AutoCAD	računari 91
MS DOS 6.0	računari 92
CorelDraw	računari 93
Moja škola paskala	računari 94
Savršeni PC	računari 95
Ventura Publisher	računari 96

3 Bankе se modernima (96/74)
4.Clipper u mraži (96/66)
5.Dejan nije kod kuće... (89/61)
8.Elementarne metode (92/58)
7 Elementarne metode (pretraživanje) (89/78)
8 FileFind za početnike (91/72)
9.Formatiranje tabela (94/66)
10 Grafični objekti (95/98)
11.Hufmanove stable (96/66)
12.Jedan pravi TSR (93/64)
13 Kako da uradim (Windows aplikaciju) (94/72)
14 Knjiga u kojoj sve piše (88/88)
15 Kemerci K'o nula (traženje grešaka) (91/70)
16 Kontrola tastature (94/86)
17 Laki uvod u DPMI (96/56)
18 Memorija i modeli (C) (93/57)
19 Memorija na klin (90/70)
20 Napredne metode (pretraživanje) (91/63)
21 O stringovima i ljuskama (94/80)
22.Objekti za nestrpljive (96/89)
23 OLE ad A do Š (92/51)
24 S druge strane ogledala (PS/TrueType) (91/58)
25 Sistemsko okruženje (environment) (90/70)
26 Slogovi i polja (Paradox Engine) (95/58)
27 Sortiranje u akciji (96/80)
28 Strogo po kalendaru (94/76)
29 Sudar svetova (96/80)
30 TSR uzvrata udarac (94/80)
31 U kom grmu leži bag? (96/56)
32.Zamenite motor ako su podaci spori (94/72)

37 Umetak
1 AutoCAD (91/ 0)
2 CorelDraw (93/ 0)
3.Microsoft Word 2.0 (89/ 0)
4.Moja škola paskala (89/ 0)
5 MS C/C++ 7.0: strukture i funkcije (96/ 0)
6 MS DOS 6.0 (92/ 0)
7 PKZIP 2.04C (89/ 0)
8.Savršeni PC (94/ 0)
9 Snažni perfv (Bil Gejts) (96/ 0)
10 Ventura Publisher (96/ 0)
11 WordPerfect for Windows (94/ 0)

38 Uslužni programi
1 Do pešačenja beše (na dlanu) (94/45)
2 Dok su drugi spavali (ARJ 2.41) (94/38)
3 Držite alku (Hijack) (94/31)
4 Gospodar desktopa (PCT for Windows) (93/50)
5 Napravi mi malo mesta (Stacker fix) (94/31)
6 PKZIP 2.04C (89/ 0)
7.Prelno hleba QEMM (7.0) (94/66)
8 Rastegljivi diskovi (Stacker) (96/82)
9 U oemaj rundi (PC Tools 6.0) (91/34)
10.Ubrzavanje diskova (93/47)
11 ZIP (2.0) je stigao (89/52)

39 Veštačka inteligencija
1 Duel (ChessMaster vs. Stalker) (96/82)

40 YU kompjuterski život
1 Bile je lepa, mogle je bolje (Olimpijada) (96/20)
2 Centrisi bežičnih jaka (96/ 9)
3 Daveti tom („Računari" 1988. godine) (96/81)
4 Hepening u Hajdu (96/ 9)
5 Imiel više od kompjutera (96/ 9)
6 Indeks tema (96/95)
7 Ipak za kreče (93/19)
8 Kvartijeri čipavi (94/77)
9 Mileni za najbolje YU programere (93/24)
10 Nauka na stranice ovde (V. Matejić) (93/20)
11 Nije nam lelike (Comtrad) (93/25)
12 OSA u Novom Sadu, Sibiru i na Kipru (96/23)
13 Poslovne knjige (89/55)
14 Računar CT386SX debio D. Stulić iz Zemuna (94/51)
15 Sesam Biten (94/88)
16 Sesam Biten (96/85)
17 Sesam Biten (96/80)
18 Sesam Biten (91/78)
19 Sesam Biten (92/79)
20 Sesam Biten (89/79)
21 Sesam Biten (96/84)
22 Sesam Biten (89/79)
23 Sesam Biten (96/80)
24 Šta ame ispali iedi sme biti mali (96/62)
25 Tihi čovek Mika (in memoriam) (96/87)
26 Uprkos svima, uprkos svemu (Sajam tehnike) (93/17)

41 Zaštita programa i podataka
1 Arghl (96/86)
2 Ži Varnav problem (96/86)

HARDVERU JE POTREBAN SOFTVER
VAMA SU POTREBNI *računari*

BEOGRAD
Kuća knjižarstva „Kultura"
Terazije 45, 011/330-906, 330-627

BEOGRAD
„Kultura"
Bulevar revolucije 239 011/418-731

BEOGRAD
„Kultura"
Kosovska 37 011/321-250

BEOGRAD
„Kultura"
Požeška 61 011/553-361

BEOGRAD
„Kultura"
Ratka Vojvoda Čola 28 011/5336-426

BEOGRAD
Petrica Lumumbe 40 011/771-894

BEOGRAD
Požeška 136 011/580-226

NOVI BEOGRAD
Narodnih heroja 11 011/601-522

ALEKSINAC
Maršala Tita 133 018/873-010

BABUŠNICA
27 jula 2 018/ 85-006

BANJA LUKA
Maršala Tita 67-a 078/ 23-912

BANJA LUKA
Maršala Tita 67-a 078/ 26-633

BAR
Jovana Tomaševića 13 085/ 27-299
27-291

BELA CRKVA
Prvog oktobra 72 013/851-039

BOR
Radnička 4 030/ 31-240

DRVAR
Maršala Tita 155 077/622-576

JAGODINA
Kneginja Milice 14 035/222-775

KNJAŽEVAC
Trg oslobođenja 10 019/ 41-330

KULA
Maršala Tita 250 025/722-442

LAZAREVAC
XIII proletarske 16 011/8124-635

JEFTINO
U KNJIŽARAMA BIGZ-a
POPUST 20%

OD SADA I U UŽICU TRG PARTIZANA 12

JEFTINIJE
U PRODAVNICI BIGZ-a
POPUST 30%

Prodavnica BIGZ-a, Bulevar vojvode Mišića 17 11000 Beograd, je otvorena svakog radnog dana od 7 – 15 sati. Dođite ili rezervišite svoj primerak na (011) 651-666, lokal 259 ili 226

NAJJEFTINIJE U PRETPLATI

10 BROJEVA (Godina dana) 30 DEM
6 BROJEVA (Pola godine) 20 DEM

Pretplata je najjeftiniji i najsigurniji način da dođete do svog primerka lista. **Najjeftiniji** zato što je danas sve jeftinije nego što će biti sutra i zato što pretplatnicima odobravamo posebne popuste. **Najsigurniji** zato što ne trebe da razmišljate da li ćete „Računare" naći u prodaji ili ne. Nudimo Vam dve mogućnosti

- DOSTAVU PREPORUČENOM POŠILJKOM
- LIČNO PREUZIMANJE U PRODAVNICI BIGZ-a

Važi prodajna cena na dan pretplate. Za uslove i detalje se raspitajte na telefon (011) 651-666 lok. 259 ili 226

Mogućnosti ima puno, a Vi se odlučite za onu koja Vam najviše odgovara. Ali, svakako se odlučite za Računare

- zato što objavljuju najnovije tehničke i poslovne novosti iz zemlje i sveta
- zato što objavljuju prikaze najnovijeg softvera i hardvera
- zato što objavljuju praktična rešenja za problema na koje nailazite u korišćenju i programiranju računara
- zato što predstavljaju najveću barzu u zemlji za kupovinu i prodaju računarske opreme
- zato što u svakom broju objavljuju dodatak na 32 strane
- zato što tekstove pišu poznati i kompetentni autori
- zato što lepo izgledaju
- zato što često nagrađuju svoje čitaoce stvarima koje su im najpotrebnije

Pobrinite se na vrema da ne ostanete bez svog primerka "Računara"!

LESKOVAC
Jašac blok 1 016/43-348

MLADENOVAC
Maršala Tita 16 011/8221-000

NIŠ
Obilićev venac 78 018/ 21-580

NIŠ
Važdova 4 018/ 22-628

NOVA VAROŠ
Rade Stevović bb 033/ 82-101

NOVI SAD
Almoška 3 021/ 51-993

OBILIĆ
KOD PRIŠTINE 026/ 61-648

PANČEVO
Braće Jovanović 22 013/ 43-474

PANČEVO
Dimitrija Tucovića 1-a 013/ 41-598

PANČEVO
Žarka Zrenjanina 3 013/46-802

PIROT
Trg AVNOJ-a bb 018/ 22-657

PODGORICA
Bulevar revolucije 40 081/ 14-910

PODGORICA
Bulevar revolucije 40 081/ 42-105

POŽAREVAC
Moše Pijade 38-40 012/223-160

POŽAREVAC
Stari korzo 3 012/223-073

PRIŠTINA
Kosovskih brigada d-1 038/26-760

PROKUPLJE
Ratka Pavlovića 8 027/ 21-006

SMEDEREVO
Kralja Petra 2 026/222-001

SURČIN
Vojvodanska 73 011/8440-245

VALJEVO
Vojvode Mišića 23 014/ 27-567

VELIKA PLANA
Voje Jeremića 9 026/ 51-932

VRANJE
Žikice Jovanovića bb 017/ 23-459

ZRENJANIN
Kralja Aleksandra 6 023/ 61-113

ŽELEZNIK
Darinke Radovic 39 011/571-434

lider

ili vredna i važna stvar

Biznis kao uspeh, vreme kao porodica, nauka kao način rada, kultura kao svet... Izgleda da se otuđujemo od vrednih stvari. U poplavi raznih lidera Mi Vam predlažemo ovog: ***Sun Microsystems*** Lidera na svetskom UNIX tržištu... I to je jedna vredna stvar. Novinski oglas je malo prostora za tako važnu stvar kao što je informatika u Vašoj instituciji, zato nas kontaktirajte jer... reč je o važnim stvarima.

Sun Microsystems

Ovo je namenski oglas. Namenjen je Vama koji o vrhunskim institucijama ovog društva odlučujete o... parama i vremenu. O vrednim stvarima.

SAGA 11070 BEOGRAD, M. POPOVIĆA 9, SAVA CE...
TEL. 011/222 3679, 147 182, 222 4323 EXT. 256 & 259,
222 4322, EXT. 256 & 259 FAX: 011/458 755, 605 576

A sada Vi Combinujte

CT 386SX -109

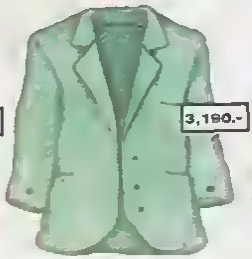
1,990.-

CPU 386SX/25 MHz
2MB RAM
Floppy 1.2 MB I 1.44 MB
Tvrdi disk 105 MB IDE
SVGA 1024 x 768 / 512 kB
SVGA kolor monitor
slimline kućište_mlđ

CT 386/40 -125

2,190.-

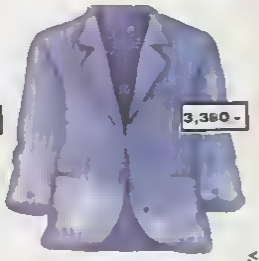
CPU 386DX/40 MHz
4MB RAM cache 128kB
Floppy 1.2 MB I 1.44 MB
Tvrdi disk 125 MB IDE
SVGA 1024 x 768 / 1 MB
SVGA kolor monitor
mini tower kućište mlđ



CT 486/33 -210

3,190.-

CPU 486DX/33 MHz
4MB RAM cache 256kB
Floppy 1.2 MB I 1.44 MB
Tvrdi disk 210 MB IDE
SVGA 1024 x 768 / 1 MB
SVGA kolor monitor
mini tower kućište mlđ



CT 486/50 -210

3,360.-

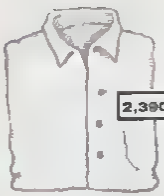
CPU 486DX/50 MHz
4MB RAM cache 256kB
Floppy 1.2 MB I 1.44 MB
Tvrdi disk 210 MB IDE
SVGA 1024 x 768 / 1 MB
SVGA kolor monitor
tower kućište mlđ

SM 586 TDH TALENATA

HEWLETT PACKARD

3,999.-

HP SCAN JET IIC
kolor skener
rezolucija 400 dpi,
A4 format
24-bitne boje



HEWLETT PACKARD

2,390.-

HP SCAN JET IIP
crno beli skener
rezolucija 300 dpi
A4 format

HEWLETT PACKARD

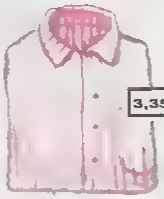
1,090.-

DESK JET 510
Ink jet štampač
rezolucija 300dpi
A4 bafer 16 kB
kvalitet ispisa kao
laserski štampači

HEWLETT PACKARD

1,800.-

HP LJ 4L
laserski štampač
rezolucija 300dpi
1MB memorije
RET tehnologija
YU slova



HEWLETT PACKARD

3,350.-

HP LJ IV
laserski štampač
rezolucija 600dpi
2MB memorije
RET tehnologija
46 fontova

CASIO

340.-

DR-8620T
16 cifara
štoni sa trakom

CASIO

200.-

FX-8700G
naučni
programabilni

CASIO

280.-

SF-8300
Digital diary

ComTrad računari su dugo tražene kombinacije savršenog sastavljanja delova u kompaktne celine.
Sada VI pronađite kombinaciju koja Vama odgovara!

Garencija 12 meseci za računare i štampače, 6 meseci za kalkulatore. U cenu štampača uključen Centronix kabl

ComTrad

Combinacija Tražena dugo
Genex apartmani, Vladimira Popovića 6,

011 - 222 41 51
011 - 222 26 52
011 - 222 41 39

"REDEK", Beograd 011 - 700 150, 784 451 • SPRINT, Novi Sad 21 - 623 737 • MANIN, Kuma 022 421 265 • INFOTRADE, Priština 038 26 630 • COMTRAD, Kragujevac 034 513 928

nastv promenljive koja se učitava, itd. Evo kako približno izgleda ekvivalent naredbe @ 2,3 GET x:

```
oGet := GetNew(2,3,
oGet:varGet := {|| x},
oGet:varPut := {|y| x := y},
oGet:name := "x",
```

Funkcija **GetNew()** kao rezultat vraća novi **GET** objekat. Za svaku **GET** naredbu kreira se po jedan **GET** objekat i svi kreirani objekti se upisuju u niz. Naredba **READ** se prepreosira u pozivu funkcije **ReadModal** (izvorni kod je dat u datoteci **GETSYS.PRG**) koja na osnovu zadatog niza objekata vrši ekranska unos podataka.

Korišćenje **GET** objekata o svakodnevnom radu nije neophodno se većinu primera navodi je develjan standardni **SAY..GET** *sistem*. Međutim, objekti pružaju veliku fleksibilnost ako vaša zatreba neki specifičan unos, **GET** objekti omogućavaju njegove brzo i efikasno kreiranje.

Objekti takođe omgu biti korisni prilikom pisanja **VALID** i **WHEN** uslova su unos podataka. Naime, pri pisanju ovih funkcija nije potrebno kao argumenti prenositi nazive promenljive ili neke drugo podatke jer sve to (i više) postoji zapisano u odgovarajućem objektu **GET** objekat koji je trenutno aktivan (dobija se funkcijom **GetActive**. Tako bi funkcija su proveru uslova su unos polola sa.

```
GET( "DATUM" )
```

Sada je **oGet** objekat koji sadrži podatke o tekućoj **GET** naredbi **oGet.name** je naziv promenljive čije vrednost se učitava **oGet.type** je tip promenljive, itd. **GET** objekti imaju i interne funkcije za se tako su **oGetvarGet()** dobije trenutna vrednost promenljive koja se učitava, a se **oGetvarPut(y)** ta vrednost se postavlja na **Y**

TABELARNI PREGLED

Tabelarni pregled podataka je, o čemu da se datoteka najpreglednije predstavljaju tabelama, izuzetno važan u Clipper programima. Tabelaru pregled se o Clipper-u realizuje preko **TBrowse** objekata. Svaki **TBrowse** objekat sadrži sve informacije koje su potrebne su tabelaru pregled šta treba prikazati u kolonama, koji su razmak među kolona, koje su boje pojedinih vrednosti itd. Veštu tih podataka treba zadati prilikom kreiranje objekta a neki su postavljeni su podrazumevana vrednosti.

Najvažnija pri definisanju tabelarnog pregleda je definisanje kolona: šta se u kojoj prikazuje, koji su naziv, i sl. Svaku kolonu se opisuje posebnim **TBColumn** objektom koji sadrži sve te podatke. Sledeće naredbe kreira jedan takav objekat

```
oColumn := TBColumnNew( "NAZIV KOLONE",
                        {|| upper(Ime)} )
```

Funkcija **TBColumnNew** kreira **TBColumn** objekat. Njeni argumenti se nalze iznad kolonu i kodni blok koji definiše šta treba prikazati u koloni.

Novi **TBrowse** objekat se kreira pozivom funkcije **TBrowseDB**

```
oTB := TBrowseDB( 0,0,24,79 )
```

Argumenti su koordinate prozora u koroe će se vršiti tabelarni pregled. Sada treba kreirnom **TBrowse** objektu saopštiti koje će su kolone tabelarnog prikaza, tj. dodati sve definisane kolone. Za to se koristi interna funkcija **addColumn** objekta **oTB**

```
oTB:addColumn(oColumn1)
oTB:addColumn(oColumn2)
...
```

Pored svoga, mogu se poslediti neki parametri. Na primer, može se specificirati da su prve dve kolone u prikazu "zamrznute"

```
oTB:freeze := 2
```

Sada je objekat **oTB** spreman za rad. Pregled se svodi na učitavanje komande od korisnika i slanje odgovarajuće poruke o objektu **oTB**. Pre svega treba naložiti objektu da ispše svoje kolone u zadatom delu ekrana.

```
oTB:forceStable()
```

Ova metoda ima naziv „forceStable" jer je moguće optimizacija vrši i postepenim korišćenjem metode **stabilize**. Sledeći korak je učitavanje komande od korisnika (na primer naredbom **Ch := Inkey(0)**) i analiza učitane komande. Upravljanje u objektom se svodi na kontrolu pomeranja kursora i slične operacije. To se radi aktiviranjem odgovarajućih internih funkcija. Na primer

```
oTB:down()
```

Znadmaja je sledeće: ukoliko je korisnik pritisnuo taster za strelicom dole, **oTB** objekta se nalaze da pomeri kursor nadole pozivom interne funkcije **down**. Opisanu sistem je pojednostavljen u realnim slučajevima treba postaviti veće više parametara i uraditi nekolike operacije više, ali najvažnije su navedene akcije. Za detalje možete pogledati originalne delu montaciju ili tekstove objavljene u „Računarima"

Nenad Batoćanin

CLIPPER 5.2

KOMPLETNO UPUTSTVO ZA UPOTREBU

SPECIJALAN ODDATAK ČASOPISA *računari*
januar 1994.

Clipper je programski jezik nastao kao jedan od mnogih *dBase* prevodilaca. U međuvremenu su se pojavili novi programi koji su uglavnome zamenili ostarili *dBase* ali je i pored prilično brzog nedostatka kao standard ostao format podataka (**DBF**) i sintaksa konstrukcija *dBase* programskog jezika. Prve verzije *Clipper*-a su imale sličnost ili čak nekompatibilnost sa kasnijim verzijama *dBase* jezika. Kasnije verzije su se sve više udaljavale od te sličnosti, tako da poslednje verzije *Clipper*-a veoma malo liče na nekadašnjeg pa čak i na *dBase*.

U to vreme prvim verzijama *Clipper* je imao nekih upotrebljivosti koje su olakšavale programiranje i koje su mu donele veliku popularnost. Poslednja verzija *Clipper* u (5.2) je savremen proizvod sa puno originalnih rešenja. Osnova *Clipper*-a je moćan programski jezik namenjen pisanju poslovnih aplikacija koje upravljaju bazama podataka. U samom svom srcu sadrži poseban dodatak koji olakšavanje pisanje takvih programa (dibager programa za izmenu baza podataka,) a postoji i veliki broj nezavisnih firmi koje nude programe namenjene razvoju programa u *Clipper*-u. Takva ponuda je izuzetno raznovrsna, počevši od kolekcionih raznih funkcija pa do *CASE* alata za automatsko projektovanje i generisanje programa.

Najvrednija (i najmanja) stvar u soft-verskom proizvodu *Clipper* je programski jezik: to je upotrebljivan programski jezik sa mnoštvom rešenjima, od kojih su neka preuzeta iz drugih programskih jezika (*dBase, C*) a neka su potpuno originalna. Neka od tih rešenja čine *Clipper* veoma zanimljivim na proučavanje.

Naredni tekst će se najvećim delom baviti *Clipper* programskim jezikom. Biće okarakterisane ove važnije konstrukcije tog jezika i načini upotrebe u programima. Međutim to ne treba shvatiti kao zamenu za originalno uputstvo koje se dobija uz program i originalnu dokumentaciju se opisuju svi detalji naredbi i funkcije dok će ovde biti reči samo o najvažnijim stvarima. Originalna dokumentacija ima se cilj da po potrebi objasni ma kakav i veoma sitan pojedinih konstrukcija, dok je u ovom tekstu naglasak na objašnjavanju principa korišćenja. To znači da ovaj tekst može da se koristi kao štivo za učenje *Clipper*-a i se razjašnjavanje pojedinih konstrukcija, dok bi dokumentacija služila pa potrebi kao referentni priručnik.

"Računari" su već objavili uvodak o *Clipper*-u (februar 1992) u kome su obrađena osnovna programiranja na *Clipper*-u. Ovaj tekst se može shvatiti i kao svojevrsan nastavak tog uvodka. Za štivanje je neophodno bar minimalno poznavanje *Clipper*-a ili nekog srodnog programskog jezika.

OSNOVNI ELEMENTI CLIPPER PROGRAMSKOG JEZIKA

Programski jezik je onoliko dobar i elegantan koliko su kvalitetne njegove osnovne konstrukcije programeri su vezu osećaju za stepa bilo koja vrste i ne vole jezika koji ih spu tavaju nasprama naredbama. Osnovna konstrukcija *Clipper*-a se se razlikuju prvide od konstrukcija u bilo kom programskom jeziku postoje promenljive tipovi podataka, naredbe funkcije, itd. Međutim postoji jedan glig više u odnosu na klasične programske jezike: mnoge stvari su ugrađene u sam jezik. To je i razumljivo, jer se uzma u obzir da je *Clipper* specijalizovan programski jezik.

TIPOVI PROMENLJIVIH U CLIPPER-U

Za razliku od većine standardnih programskih jezika, *Clipper* spada u takozvane **netipširane** programske jezike. To znači da promenljive nemaju uvek vrednost koja pripada samo jednom tipu podataka, već svaka promenljiva može imati vrednost različitih tipova. Na primer:

```
X := 100
X := "ABC"
X := Date()
```

U ovom primeru promenljiva **X** redom dobija vrednosti koje su tipa broj, string i datum. Kada se dodeli nova vrednost, automatski se poništava prethodna i leda je tip promenljive zadnje zatvam druge.

Da bi se proverile koj je tipa promenljiva, koriste se funkcije **ValType** i **Type**. Funkcija **Type** se zadati naziv promenljive vrade simbol koji označave njen tip. Na primer:

```
Num := 100
? Type("Num")      // "N"
```

Type kao argument prihvata i niza, ali treba imati u vidu da niz ma koji elemenat može imati različit tip. Funkcija **Type** ne pripada biblioteci **CLIPPER.LIB** nego biti dobro analizirati pa će rezultat biti "U" u slučaju se funkciju **Type** takodi se ne mogu koristiti promenljive klase **STATIC** i **LOCAL**, što je veliki nedostatak.

Kao zamena za **Type**, može se koristiti funkcija **ValType**, koja radi slično pesne. Funkcija **ValType** treba proslediti niz čiji će odredbenjen, ali dodavse zadat o na kao string. Prethodni primer bi sada bio:

```
LOCAL Num := 123
? ValType (Num)    // "N"
```

Ovakva osobina promenljivih ima svojih loših i dobrih strana. Glavni nedostatak je povećana mogućnost pojave grešaka u programu. Na primer:

```
X := "123"
? X/2
```

Ova naredba će prevodilac prevesti bez ikakvih problema. Međutim, greška je očigledna. U drugoj odredbi se pokušava dedjenje stringa brojem, što rezultuje greškom. Kako prevodilac na vodi evidenciju o tipovima promenljivih on nema nikakvu informaciju o tipu promenljive **X** u trenutku prevođenja, pa sete i nastaje ovakva greška.

S druge strane ova osobina *Clipper*-a može biti veoma korisna. Na primer, ako imate neku operaciju koju želite da primenjujete na promenljive raznih tipova, ali potrebna da se svaki tip podataka pošete odvojenu proceduru već je dovoljno napisati zajednicu proceduru. Kada se ona obrada izvršiti, može se odrediti na osnovu funkcije **Type** ili **ValType**. Mnoga u gradene *Clipper*-ove funkcije koriste ova osobinu. Na primer takva je funkcija **PadR** koja služi za popunjavanje podatka blanko simbolima sa desne strane. Ako se kao argument **PadR** zada string rezultat je string popunjen da zadate širine:

```
PadR("ABC", 5)                    // "ABC  "
```

Međutim kao argument se može zadati i broj ili datum.

```
PadR(100, 5)                      // "100  "
PadR(Date(), 8)                   // "21 06 92"
```

Pisanje ovakvih funkcija je veoma jednostavno: funkcijom **ValType** treba ispitati koj je tipa argument koji je prosleden i u zavisnosti od toga preduzeti odgovarajuću akciju.

```
PROC Test (x)
  IF ValType(x) == "N"
    // Obrada za brojeve
  ELSEIF ValType(x) == "D"
    // Obrada za stringove
  ...
ENDIF
RETURN
```

Zaključak je da nepripadnost promenljivih povećava mogućnosti pri programiranju ali se time povećava i mogućnost greške usled upotrebe neodgovarajućeg tipa podataka. Zato ova osobina treba pažljivo koristiti - u većini slučajeva može biti korisna, ali ponekad može biti i štetna.

NAZIVI PROMENLJIVIH I FUNKCIJA

Možda će vam se učiniti da je izbor promenljive veoma važna svim sporednim i nevažnim u programiranju. Take se u programima pišanim „na brzinu" veoma često srešu „majstorije" nazivi poput **p, q**, i slično. Vrsna koje dobijete na skraćenanu analizanji imena promenljivih ili funkcija svakako često kamnije izgubiti i to više puta, prilikom analize svog programa kada budete pesavo pokušavali da odgovorite sušene davno napisano procedure ili budu pokušavate da se sete kako se zove vaša funkcija koju ste napisali za određenu operaciju. U suštini, nazivi funkcija i promenljivih ne sušetne važnu, bar utoliko koliko i komentari u programu.

U principu *Clipper* ne pravi razliku između malih i velikih slove pri prevođenju programa rachka je jedino važna zbog preglednosti. Nazivi funkcija se mogu pisati na više načina. velikim slovima, malim slovima ili kombinacijom. Najčešći slučaj je kombinacija velikih i malih slova, a ima što nazivi počinju velikim slovom i velikim slovima su označene posebne celina u nazivu. Na primer **SetColor**, **PrecLine** i **ol**. Koji ćete sistem izabrati nije pravilo važno najvažnije je da se tog ni atema držite. Ta će vam omogućiti da se u vašim programima snalazite kao kod kuće na prvi pogled čete uočavati šta su promenljive ključne reči, funkcije. Takođe je važno da nazivi funkcija ne budu kriptografski i da asocirajo na operaciju koju te funkcija obavlja. Bez obzira što nazivi može biti proizvoljne dužine *Clipper* ustima u obzir samo prvih 10 ima bela.

Pošto *Clipper* ima promenljive koje mogu biti raznih tipova u raznim delovima programa, dobre je da nazivi promenljivih budu prilagođeni toj osobini. U dokumentaciji *Clipper*-a se se nazive promenljivih predlaže jedna varijanta tzv „mađar-

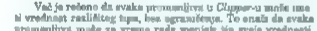
ske notacija" u kojoj prefiks svake promenljive određuje njen tip i značenje. Tip prefiksa je za krajnjeg e za skraćene d ee datume i sl. Ovaj sistem imena se pile je lakše pravilima kao i na zv funkcija. Na primer:

Sada su promenljiva i njihov tip vezana nedjivi. Tako se sprečavaju greške tipa **cXXXnYYY** - sve se lako može da se izbegnu neodgovarajući! Problem ostaje kod promenljivih čiji tip nije unapred poznat. tada se koc prefiks ukratko koristi x. Na primer: **oVar** i sl.

Za imena se koristi prefiks a, s tim što se obično dodaje na prefiks se tip elemenata niza.

Sada o označava da se radi o nizu, a za da su elementi brojevi. Međutim, ono je redak složaj nizovi z *Clipper*-u nadrže veoma raznorodne podatke, tako da prefiks se tip elemente tada gubi smisao. Zato se obično upotrebljava samo prefiks a.

Originalna mađarska notacija nije zamišljena suviše strogo i predviđa proširenje prefiksa još nekim informacijama o značenju promenljive. Tu značenje promenu u složaju polja u DBF datotekama. Na primer se polja se u prefiks može dodati slovo f:



Ipak, dalase raširenji sostad je korišćenje prefiksa koji označava datoteku kojoj pripada to polje. Na primer, za datoteku **ARTIKLI** naziv polja bi mogao biti **A_NAZIV**. Ni jedan od ova dva načina nema nake određene prednosti, tako da možete izabrati onaj koji vam više odgovara.

Što se tiče tipova promenljivih, relativno često je potrebno ra teski način razlikovati globalne promenljive (**PUBLIC**) od ostalih. Ove se jednostavno može urediti tako što uvedete slove p u prefiks. Na primer: **npOvi**.

Treba imati na umu da se ne može sprhanume stvari i da de program lete rediti ma kako osmišljeni promenljive i fun koje – radi se samo o tekućoj fazi. U svakom slučaju, vema je preporučljivo da izvojite jedne noted i da ga striktno poštujete o svim programima

MEMORIJSKE KLASE

Već je rečeno da svaka promenljiva u *Clipper*-u može ima bi vrednost različitog tipa, bez ograničenja. To znači da svaka promenljiva može za vreme rada menjati tip svoje vrednosti. Svaka promenljiva osim toga ima i jedna zapremnijeru karakteristiku koje zavisi od toga kojoj **memorijskoj klasi** pripada. Od izabrane memorijske klase potpuno zavisi penalazje i vidljivost promenljive u odnosu na procedure i funkcije programa. Sve *Clipper* promenljive mogu se grubo podeliti u dve memorijske klase.

Prva memorijska klasa: tzv. **LOCAL** i **STATIC** promenljive. Ove promenljive se potpuno definišu za vreme prevođenja programa. Drugu memorijsku klasu čine **PRIVATE** i **PUBLIC** promenljive koje se kreiraju po potrebi se vreme i izvršavanje programa. Ovde se svakve dinamičke promenljive moraju kreirati pre upotrebe pritom i rad se njima je dosta kompleksnanji. Zato je vrlo koljo koristiti **LOCAL** i **STATIC** promenljive

LOCAL promenljive se definišu se nemo prevođenja programa i kreiraju se po ulasku u proceduru u kojoj su definisane. Vidljive su samo u toj procedori ni jedna procedure ili funkcija koja se pozava a ta procedura ne može pristupiti direktno lokalnoj promenljivoj. Promenljiva tipa **LOCAL** se može deklarisati na početku modula naredbom **LOCAL <lista imena>**. Ove promenljive treba najviše koristiti um spu njavajuć unlov potrnosg „skrivanja informacija" koji bi trebalo da zadovoljava svaka procedura u programu

U ovom primeru se deklarisku dve lokalne promenljive **X** i **D** i istovremeno se inicijalizuju. Procedura **PrmPS** na može prizvati ni jednaj od nje dve prouzrokuje. Inicijalizacija promenljivih se obavlja uvek po ulasku u proced ri i **DemoLocal**. Iz nadi da do se **X** i **D** uvek isteva postavljati na lucijalna vrednost kada se aktivira procedura **DemoLocal**. Primetite da se u inicijalizaciji mogu koristiti i prstvoljni izrazi ne samo konstante. Take do prethodni primer uvek lepisati broj 1 i tekuće datume

Promenljive klase **STATIC** se takođe definišu pri prevođenju, ali se se razliku od **LOCAL** promenljivih ne inicijalizu ju uvek kad pođne izvršavanje funkcije u kojoj se definisane Deklarišu se naredbom **STATIC <lista imena>**

Promenljiva **X** zada postoji i ščva prethodna vrednost i Sada prevođenja **DemoStatic** nije aktivirana. Inicijalizacije se vrši samo jednom i to pri prvom pozivu izvršavanje. Sada se kao ranije se ia cijalizuju samoiji upotrebiti konstantni izrazi Prethodni primer be pri svakom pozivu ispisati **X** uvećan za 1 1, 2, 3 ovakve promenljive su vidljive samo u modulu u kome su definisane: ništa lokalnim procedure koje se pozivaju „na više" promenljive tipa **STATIC**

Nešto se različuje složaj ako se promenljiva deklariše kao **STATIC** na početku datoteke izvan svih procedure i tada se vidljive u svim procedurama u toj datoteci jako pogodno se „interna" globalna promenljiva na uvou jednog modula. Naredbe **STATIC** i **LOCAL** se moraju pisati pre svih ovršnih naredbi u modulu u kome se nalaze se razliku od složaja na realu kao što je **PRIVATE**, ove su samo deklarativne a ne i zvršne naredbe

Dinamičke promenljive se kreiraju izvršavanjem naredbi **PUBLIC**, **PRIVATE** ili naredbom dodjljivanja za vreme i zvršavanja programa Globalne promenljive se kreiraju se vreme izvršavanja programa naredbom **PUBLIC** i takve promenljive su vidljive iz svih procedure u programu Promenljiva vi tipa **PRIVATE** se takođe kreiraju za vreme izvršenje programa, e vidljive su u proceduri u kojoj su definisane i u svim procedurama koje se pozivaju iz te procedure Ako se u nekoj proceduri navedu naredba dodjljivanja **X := <izraz>** a pri tom **X** uje prethodno definisana, automatski će se sformirati promenljive **X** tipa **PRIVATE**. Ove je korisna sposobnost koje vam omogućuje da pravite inusne fleksibilne programe ali krije u sebi dosta opasnosti Jedna od osnovnih je smanjena brzina izvršavanje programa

U prethodnom primeru, u proceduri **DemoDyn** promenljive **prX** je vidljiva i procedurama **DemoDyn** i **PP** e promenljive **pnY** u svim procedurama u programu Promenljive **prZ** i **prW** (automatski se kreire naredbom dodjljivanja) su vidljive samo u proceduri **PP** i u proceduranma koje ona poziva

Osim što je red se ovakvim promenljivim bitno opreza u odnosu na **LOCAL** i **STATIC** promenljive vrlo je lakko odr-

šak nesvjesnost pojavljih procedura od ostatka programa. Nikako šta vrši ako je procedura napisana bez greških, kad sve procedure koje ona poziva mogu neometano da rezvijaju sve promenljiva klase **PRIVATE** u ovoj proceduri – to znači da pregledom procedure na možete reći da li će ispravno raditi ili će nastate tvrdnji tak kad pregledate i sva podredane procedure!

Bez obzira na iste osobine, promenljive klase **PRIVATE** se relativno često koriste, bol zbog svoje vidljivosti u podredenim procedurama i funkcijama. na taj način se ostvaruje „lokalna globalnost" promenljivih vesma pogodna u nekim situacijama. Tada je preporučljivo koristiti opciju /w pri prevođenju programa Time se postiže da prevodilac prijavljuje sve neiskliarisane promenljive koje se upotrebljavaju. Da bi se izbegle upozorenje se dinamičke promenljive, potrebno je pepknom deklaracijom **MEMVAR** naznačiti sve dinamičke promenljive koje će biti korišćene u tej proceduri.

U ovom primeru su deklarisane promenljive **X** i **Y** kao dinamične se koriščene u proceduri **Test**. Pri tome se samo **X** kreira u proceduri **Test** - promenljive **Y** je vidljive iz neke „spoljne" procedure. Deklaraciju **MEMVAR** možete upotrebljavati i ispravi svih procedura u datoteci i one tada ima važnost za sve procedure i funkcije u datoteci.

Važna je da uočite razliku između neke sličnih **LOCAL** i **PRIVATE** promenljivih. Na primer, nije dobro mešati deklaraciju **PRIVATE** u **LOCAL** bez dodatne provere, jer su promenljive klase **PRIVATE** vidljive u tekućoj proceduri i u svim koje pozva tekuće e klase **LOCAL** samo u tekućem modulu. pa vrlo se u tom slučaju može desiti da dobijete poruku o nedeklarisanoj promenljivoj ili još gora, da se kreira nova promenljiva koja se originalnim nema nikakve veze Ova osobina promenljivih klase **PRIVATE**, mada je u principu suprotna pravilima dobrog programiranja, vesma se često koristi. To se vidi u sledećem primeru

Promenljiva **X** u proceduri **Main** dobiće vrednost 2, jer je vidljiva i u proceduri **SubRoutine**. Međutim, ako se **X** deklariše ke kao **LOCAL** biće vidljiva samo u proceduri **Main**, pa će procedure **SubRoutine** automatski kreirati novu promenljivu **X** klase **PRIVATE** koje je samvim samostima od promenljive **X** u proceduri **Main**.

PRIVATE promenljive deklarisana u glavnom programu ima potpuna iste osobine kao i **PUBLIC** promenljive, tj. u glavnom programu je svejedno da li koristite naredbu **PUBLIC** ili **PRIVATE** za deklaraciju. Međutim, u ostalim procedurama je to vesma različite ako se deklarišu promenljiva klase **PRIVATE**, sve procedure koje se „zovu" neće videti deklarisanu promenljivu. Ako se upotrebi naredba **PUBLIC** ova procedura će videti svu novu promenljivu.

Treba imati na umu da su **PUBLIC** i **PRIVATE** izvršne naredbe tako da možete pisati nekte popul.

LOCAL i **STATIC** naredbe nisu izvršne naredbe i mogu se pisati pre svih izvršnih naredbi

NIZOVI

Osim standardnih tipova podataka, *Clipper* poznaje i ni

seve Za razliku od standardnih programskih jezika, *Clipper* nizovi mogu sadržati elemente različitih tipova, pa čak i druge nizove Svaki niz je ograničen na najviše 4096 elemenata Ni zovi se mogu definisati na više načina. Najjednostavniji način je navođenje njegovih elemenata. Na primer:

Ova naredba kreira niz A koji sadrži tri elementa. brojeva 1, 2 i 3. Pošto elementi nizova mogu biti vrednosti različitih tipova, može se pisati i

Prvi niz kao elemente sma stringove, kod drugog niso je prvi element broj, drugi string, a treći tekući datum. Kada se kao u ovom primeru zada element niza kao izraz, prvo se izvrši računanje izraza i dobijena vrednost se dodeljuje odgovarajućem elementu niza. U vrašene primeru se definiše niz od dva elementa. pri čemu je prvi element broj, a drugi element je niz koji sve dva elementa. I ovaj niz dalje može imati svoje podnizove itd. do proizvoljne dubina.

Drugi način za kreiranje nizova je pomoću deklarisanje promenljivih. Tada treba navesti maksimalne vrednosti indekse u zagradi. Na primer naredba

definiše niz **X** koji će imati 10 elemenata. U početku svi ti elementi imaju vrednost **NIL**. Ovako se svaju deklarisati i nizovi sa više dimenzija. dovoljna je navesti maksimalne vrednosti svih indekse. Na primer

kreire matrim od 10x8= 80 elemenata. Broj dimenzija po nizu nije ograničen. Primetite da se ovakvim naredbama mogu kreirati samo nizovi „pravilne" dimenzija. nizovi koji kao elementi sadrže podnizove raznih dužina moraju se kreirati navođenjem elemenata ih na neki drugi način.

Elementima nizova se pristupa na uobičajen način. treba navesti naziv niza i u zagradi odgovarajuć indeks elementa. Na primer: **z[5]** označava treći elemenat niza **X**, **aNiz[3,5]** elemant matrice **aNiz** i sl. Za nizove se više dimenzije postoje dva načina obeležavanja. umesto **aNiz[4,5]** može se pisati i **aNiz[3][5]** Oba načina se potpuno ekvivalentna i zvojedno ja koji koristite Elemenu niza se mogu koristiti bez kao i sve druge promenljive.

Da bi se pravilno koristili nizovi z *Clipper*-u, treba znati kako se oni raspoređuju. Naime, nizovi u *Clipper*-u su realizovani preko pokazivača. na je obična promenljiva koja kao vrednost sadrži pokazivač na niz (stvarne elemente niza) Take se red na nizovima svodi na rad sa pokazivačima.

Naredba **Y := X** na kopira niz, već samo pokazivač na niz Take će posle ovih naredbi i **X** i **Y** sadržati pokazivače na isti objekat. na **{1, 2, 3}** Izraz poput **x[3] i y[3]** davaće iste vrednosti – u stvari radi se o **različi** istim lokacijama. Ovo se lako može preveriti ako se na prethodnu primer dvela naredba

Take će drugi elemenat niza postati 22 sada je niz **{1, 22, 3}** Posle i **Y** „pokazuje" na isti niz i **Y[2]** će imati vrednost 22 Ne ovo se relativno lake zaboravijo i zato uvek treba imati na umu šta se u stvari dešava kada radite se nizovima.

Operator poređenja **= =** radi i na nizove, o tmo što se na preverevaju vrednosti elemenata niza nego samo vrednost pokazivača. **z = = y** će biti tačno samo ako i **X** i **Y** sadrže pokazi vače na isti niz u memoriji što će se desiti pasle naredbe **z := y** U suprotnom izraz nije tačan.

```
// TRUE
// FALSE
```

Tako je druga rama netačan, bez obzira što su vrednosti u svim nizovima iste: rači se o dve kopije s operator == vraća vrednost .F..

Sada je jasno kako se realizuju nizovi koji kao elemente imaju nizove: takav element niza s stvari sadrži pokazivač na podniz. Nizovi se više dimenzija su realizovani na isti način, kod matrica je vraća element niza novi niz.

Poslednji je potrebno ručno iskoprati na. Treba se treba koristiti naredbi = – poito ona pravi samo kopiju pokazivača, a ne niza. Funkcija **ACIcna** fizički kopira niz ako je potrebno inesi pravu kopiju onda treba upotrebiti ovu funkciju.

```
* := {1, 2}
:= Aclena (x)
  := 99
[1] := 1
[2] := 2
```

Osim funkcija **ACIsave** *Clipper* sadrži još funkcija koja rada sa nizovima. No primer funkcija **Array** kreira praan niz sa zadatim brojem elemenata.

```
                    NIL
```

Kreirani niz ima vrednost **NIL** kao sve elemente. Od značajnih još funkcija treba pomenuti **AAdd** za dodavanje novog elemenia u postojeći niz, **ACopy** za kopiranje dela niza u drugi niz **ASize** za izmenu dimenzija niza, **AFill** za popunjavanje niza nekom vrednošću i sl

```
aNiz := {}          // aNiz := {}
AAdd (aNiz, "A")    // aNiz := {"A"}
AAdd (aNiz, "B")    // aNiz := {"A", "B"}
AAdd (aNiz, "C")    // aNiz := {"A", "B", "C"}
ASize (aNiz, 2)     // aNiz := {"A", "B"}
AFill (aNiz, "X")   // aNiz := {"X", "X"}
```

Od značajnijih funkcija koje služe za rad sa nizovima treba pomenuti **Len** koje vraća broj elemenata niza. Pri tome se se računa broj elemenata u podnizovima (ako ih ima) već samo broj elemenata „prvog nivoa" Na primer

```
{"A", {"B", "C"}, "D"}
```

Pasta je niz u stvari pokora promenljiva, ona može pripadati jednoj memorijskoj klasi. Sve tako može biti deklarisano kao **STATIC LOCAL, PRIVATE** ili **PUBLIC** promenljiva. Ova pravila koje važe se sličine promenljive (vidljivost, vek trajanja, ..) potpuno analogno važe i za elemente niza

S obzirom da se operacije sa nizovima aktivnjaju dinamički treba imati u vidu da se te operacije u mnogi dosta opora i se treba ih koristiti po svaku cenu. Sa druge strane nizovi u *Clipper*-u su veoma fleksibilni i to svakako treba maksimalno iskoristiti. Nizovi se, slično ostalim tipovima podataka, ravnopravno mogu koristiti u izrazima.

```
                    {"Array(4)[2]", "Array(4)[1]"}
                    {1, 2, 3}[1]
```

Nizovi se mogu prosleđivati kao argumenti procedurama i fukcijama, a funkcije kao rezultat mogu vraćati i nizove. Pri mer takve funkcije je **Array** ACIcna i sl.

MAKRO OPERATOR

Makro operator u *Clipper*-u služi se *run time* interpretiranje izraza (u dokumentaciji se govori o prevođenju) Kao argumenl makro operatora zadaje se promenljive (ili ceo izraz) lipa string. Sadržaj tog stringa onda biti sintaksno ispravan i sraz koji se zatim interpretira, odnosno računa Rezultat je vrednost tog izraza. Na primer

```
                    "A"
                    "B"
                    "A" + "B"
```

Pasle inicijalizacije promenljivih **X, Y** i **Z** vrednost **"(X+Y)*Z"** upisuje se u string **S** . ove su te navim standardne konstrukcija Međutim, već sledeća naredba prodstavlja nešto nove tu se privremeno makro operatora računa izraz sadržan u stringu **S** , znači izraz **(X+Y)*Z** Ustvaju se takvide vrednosti odgovarajućih promenljivih | računa (1+2)*3= 9 Makro operator **&** može se primenjivati ili na promenljivu koja sadrži izraz u obliku stringa (kao u prethodnom primeru) ili na izraz koji kao rezultat vraća string-izraz U tom slučaju treba izraz zatvoriti u zagrade Na primer umesto **&S** može se pisati i **&("(X+Y)*Z")** Naravno promenljiva **S** mora imati neki vrsta sat kao i izraz da bi se dobio neki efekat.

```
                    "Vrednost je: " + "10"
```

Ako sad na primer karisnik unese izraz "12 + 34" kao rezultata vrednost izraza – 46 Ako se u izrazu koriste i neke promenljive klase **PRIVATE** ili **PUBLIC** bide izračunata karakina vrednost makro operator pri interpretaciji uzima u obzir sve promenljive i funkcije koje su vidljive je takvide prosdinje. Jedini izuzetak su promenljive klase **STATIC** i **LOCAL** – ove su nevidljive za makro interpreter

Postavlja se pitanja, šta tko korisnik greškom otkuca sin takvom raliprevan| izraz? Polto je vrednost izraza nepoznata u vreme prevođenja programe ,prevodilac to ne može ni na koji način kontrolisati Za takve slučajeve je zaduđen sistem za obradu *run time* grešaka ak) se nikuce "(1+3" program će se prekinuti | biće prijavljena sintaksna greška Naravno nuda je nasbiino da jedan program prijavljuje **sintaksnu** grešku se vreme izvršavanja programa ali drugaj načina nema. Sreću, svakve (s | ostale) greške se mogu takn kontrolisati funkcijom **ErrorBlock** se kontrola se standardne procedure za obradu grešaka kako nasarave na način koga može jednostavne ispisati upozorenje i normalno nastaviti rad

Makro operator se može postvati i rekurzivne kao u sledeem primeru

```
                    "&string"
                    "&string"
                    "string"
                    "KRAJ"
```

Ovde se u stvari, računa **&("&("&("KRAJ")")")**, pa se kao rezultat štampa string **"KRAJ"** Ilez obzire na ovakvu mogućnost, treba je upotrebljavati umarene zbog dinamičkog aleksrada memorije prilikom svakovih uraknavanja.

Jedna primena definitivne rasdvaje makro operator od jednostavne funkcije se interpretiranje izraza. Naime, pomoću makroa se mogu menjati i delovi naših naredbi. Naravna u tome postoje ograničenja, ali ona su sasvim jasna prevodilac mora pri prevođenju znati o kojoj se naredbi radi da bi uopšte mogao da generiše kod, a delovi se mogu razrešiti se vreme i zvršavanja programa. Na primer sasvim je ispalna

```
                    "Alfa"
                    "Beta"
                    "Alfa" + "Beta"
```

U ovom primeru se makro operater primenjuje na string **X + "01" = "Alfa01"** : tako se kreira promenljive klase **PRIVATE** i dodeljuje joj se vrednost 12 5.

Sasvim sličan je mogućn poevate funkcije ili procedaru čije ime nije unapred poznate Funkcija se poziva kad je deo i uraza na koji se primenjuje makro operator Procedure se mogu i ovako pozvati.

```
                    "Test"
                    "Test"
```

U ovom primera sarianik izmenva mesto procedure koju treba izvršiti, ako je vrednost promenljive **Test** = .T.

KODNI BLOKOVI

Kodni blokovi (**Code Blocks**) su delovi izvrišnog kodn se kojima se mogu izvoditi odredene operacije kao se podnašina. Kodni blok se može dodeljivati promenljivej kao i vrednosti ostalih tipova podataka. Kodni blok je u stvari, funkcija bez naziva koja se može na izvestan način tretirati i kao podatak. Kodni blok je na primer

```
                    {|| Date()}
```

Ovaj kodni blok ima dva argumenta koji su ekvivalentni argumentima funkcije (X i Y) , a računa zbir tih argumenata. Umesto jednog tako kodnog bloka (ose poala "||") može sadržah ttlav niz izraza koji se uradnavaju. Na primer

```
                    {|x,y| aVec := {x, y, x+y}}
```

Ovaj kodni blok štampa zbir **X+Y**, a zatim kao rezultat vraća razliku **X Y** Rezultat je uvek vrednost poslednjeg izraza u nizu. Veliki broj naredbi *Clipper*-a je, u stvari, realizovan preko funkcija, pa se u tolu kodnog bloka mogu nalaziti proizvoljne akcija. Ako u kodnom bloku fallta da upotrebite neka naredba za koju na postoji odgovarajuća funkcija, lako ova problem rešiti pomoćnu pomoćne funkcije koja sadrži name tu naredbu

Polte je kodni blok podatak, može se dodaliti promenljivoj

```
                    {|| Date() + 1}
```

Ova naredba ništa na izračunava – ona samo dodeljuje promenljivoj **cBlk** kodni blok, ali ga pri tom na izvršava. Promenljiva koja sadrže kodne blokove na razlikuju se od ostalih i njih možete poredti, upotrebljavati u izrazima proslediviti procedurama | funkcijama, itd. Izvršavanje kodnog bloka se ekaplicitno zahteva pomoću funkcija **Eval** Ova funkcija se zadati kodni blok i nis argumenata prunanjuje kodni blok na argumente i vraća izračunatu vrednost. No primer ako se napiše

```
                    {|x| x^2}
```

biće odštampana vrednost 26 Ako kodni blok zahteva više argumenata, njih treba navesti u nastavku. Na primer– može se pisati

```
                    {|x,y| x^2 + y^2}
```

Prauda se ekoro sve šta mogu kodni blokovi može postići korišćenjem makroa, treba imati u vidu da je svak mnogo bolje koristiti kodna blokova, kad je to moguće, jer se znatno lakše testiraju (ili provode kako više u dokumentaciji) i izvršavaju se vreme izvršavanja programa, dok se kodni blokovi provode se vreme prevođenja programa, pa se kasnije se oni izvršavaju šte je mnogo brže Za dobar testiraja sledeće provodare na vašem računaru procedure **CodeBlks** će rasbiti najmanje pet-šest puta brže, se potpuno isti efekat.

```
PROC Mandlbr
  x := y := -.5 := 1
  c := "((x*y)**2)/(x y)"
  FOR i := 1 TO 1800
    x := &s
  NEXT
RETURN
PROC CodeBlks
  x := y := z := 1
  p := {|| ((x*y)**2)/(x y)}
  FOR i := 1 TO 1800
    x := Eval (c)
  NEXT
RETURN
```

Vrlo često se zaboravlja da je kodni blok, u stvari, jedan tip podataka, pa se nad njima može vršiti obrada kao i se sva kim podacima. Dodaše neke operacije nemaju smisla (nisu ni implementirane), ali jedna svakako zaslužuje pažnju kodni blokovi se mogu proslediviti kao argumenti i funkcije mogu vraćati kao svoju vrednost kodne blokova. Na primer može se pisati

```
FUNC Action (bCde)
  IF bCde == NIL
    Per := {|| 0 < Date()}
  ELSE
    Per := {|| d > Date()}
  END IF
RETURN Per
```

Postavlja se pitanje gde uopšte promenljivati kodne blokove? Priručnici ne daju baš neke ilustrativne primere za tom pitanju Vaino je zapamtiti da su kodni blokovi jedan od najvažnijih elemenata novog *Clipper*-a. Primera radi, nam mesta gde se uvek upotrebljavaju: kao šte su zadavanje akcija za objekte i procedura se obradu grešaka, ali se kasnije na još puno mesta. Naime, puno naredbi se pretprocesira i kao rezultat daje neku konstrukciju koja sadrži i kodni blok. Pogledajte definicije GET sistema i naredbe poput **COUNT** – sve je uredno vezano jednostavne | prirodno

Već smo videli da se kodni blok izvršava tek kodn se na njega primeni funkcije **Eval** To omogućuje da se drugej procedari prosledi ne samo podatak, već i neka **akcija**. Na izvestan način kodni blok je dac procedure u koju je definisan – sve statičke | lokalne promenljive vidljive u toj procedari, vidljive su i u njemu se vreme izračunavanja, bez obzira u kojej procedari se izvršava. To omogućava da se prave svojevrsni „kanali" za komunikaciju između procedura, bez globalnih promenljivih i parametara. U ovom programa procedure **Main** i **PP** komuniciraju na taj način. zahvaljujući kodnom bloku **cBlk**, lokalne promenljive **X** i **Y** iz procedure **Main** su dostupne i procedari **PP** nuda između niru vidljive u toj procedari.

```
PROC Main
  LOCAL x := 1, y := 2
  cBlk := {|| x + y}
  PP ()
RETURN
PROC PP
  ? Eval (cBlk)
RETURN
```

Osim preko funkcije **Eval** kodni blokovi se mogu izvršavati i korišćenjem funkcija **aEval** i **dbEval** Funkcija **aEval** izvrša va zadati kodni blok pojedinačno za svaki element niza čk njegovog zadatog dale Na primer – sabiranje svih elemenata niza jednostavno se može ovako napisati

```
                    {1, 2, 3, 4, 5}
                    0
                    {|x,y| y := y + x}
```

Funkcija **dbEval** se analogna koruti ona izvršave zadati kod ni blok za svaki slog aktivne datoteke O lome će kasnije biti više reči.

U funkciji **aEval** mogu se zadati opcioni parametri koji o-

znašavaju početni i krajnji element seet na kaji treba primeniti dodni blok.

Interesantna je kombinacija kodnik blokova i makroa. Često se dešt da je potrebno makro izvršavati u petlji. Tada se svaki put izvora značenja stvara sadržan u sadržene stringu, što naročito u slučaju kompliknvanijeg izraza može pokrajati. Navedja bi bila mnogo manja da se to ne radi o petlji – u tom slučaju u svakom prolazu će se izvava vršiti „prevođenje". Nevavne kodni blokje mnogo bolje upotrebiti jer se tada na vrš prevođenju, ali ima situacija kada ovaa koji treba sračunavati nije unapred poznat, već se dobija kao argument, na primer Tada se može primeniti kodni blok koji u sebi sadrži poziv na kru operatora, pomoću čuvina značenja, ali vrlo složenoma.

```
PROC Primer(XVal)
  LOCAL Rez := {}, i, Kvor
  FOR i := 1 TO Kvor TO Kvo
    Eval( {|| Rez[i] := i^XVal} )
  NEXT
RETURN
```

Ovim se ilustrovana stalna „razvijanja" makro operatora u petlji, to se radi samo na početku a zatim se u petlji izvršava prvod – kodni blok.

Mogućn je i druga kombinacija – primena makro operatora na string koji sadrži kodni blok. Na primer

```
&("X := X + 1")
Eval( {|| } )
```

Znači, string sadrži „opis" kodnog bloka, što makro operator prevodi u kodni blok, a ovaj se izvršava funkcijom **Eval**.

PROCEDURE I FUNKCIJE

Clipper na pravi razliku između procedura i funkcija, bez obzira što postoje deklaracije **PROCEDURE** i **FUNCTION** (ove se može skraćeno pisati kao **PROC** i **FUNC**) ovakve procedure se može koristiti i kao funkcija i obrnuto. Zato se u ovom tekstu termini „procedura" i „funkcija" koriste kao sinonimi.

Ipak, dobro je deklarisaga **PROC** i **FUNC** koristiti u njihovom uobičajenom značaju, procedura nema povratne vrednosti, a funkcija ima. Tako se izbegavaju eventualne kod kasnije analize programa. Naki programeri i procedure koriste kao funkcije, a tim što procedure uvek vraćaju vrednost **NIL**. Sledeći primer ilustruje ovnse procedura i funkcija u *Clipper*-u:

```
PROC Main
  ? PP(1)
  ? PP(2)
  ? FF()
RETURN
PROC PP
RETURN 123
FUNC FF
RETURN "ABC"
```

Procedura **PP** i funkcija **FF** se potpuno ekvivalentno ponašaju mogu se koristi kao procedure i kao funkcije. Ako se koristi kao procedura povratna vrednost se ignoriše

Procedure i funkcija mogu imati i parametre: to su vrednosti koje se prosleđuju iz glavnog programa odnosno se procedure u kojoj je uneđen poziv. Pri deklaraciji se ji parametri navode u zagradama iza naziva funkcije Na primer

```
FUNC Test(x, y, z)
```

Ukoliko funkcija nema parametre, treba iskoristiti umesto njih zagradama a mogu se pisati i samo zagrade Ovakva funkcija se poziva navođenjem njenog imena i lista izraza koji definišu vrednosti ulaznih parametara. Na primer, **Test (1, Date() + 7, "ABC")**

Postoji još jedan način za prenos parametara iz glavnog programa. **PARAMETERS** naredba. Ova naredba prihvata prosleđene parametre potpuno isto kao u prethodnom slučaju, a ima šta postoji jedna značajna razlika. kada se koristi prvi način (niv argument lista) prosleđeni parametri se ponašaju kao promenljive klase **LOCAL**, jak su u drugom slučaju klase **PRIVATE**. Ne programeru je da odluči koji će pronos koristiti ali treba imati u vidu da je uvek bolje rešenje koristiti pronos preko argumenta, što, bolje kao što je bolje koristiti **LOCAL** promenljive.

Pri pisanju programa veoma je važno znati nešto na koji *Clipper* prenosi parametre u procedure i funkcije Podrazumevani način je prenos po vrednosti To znači da se pre poziva funkcija prvo kopija svih zadatih parametara koje se stavljaju na stek i zatim se izvršava funkcija. Funkcija vrednosti parametara sita se staka, tako da radi se kopijama originalnih vrednosti Ts znači da izmena parametra funkcija neće uticati vas povratne funkcije menja se samo kopija, a ne i originalna vrednost. Evo primera ta ovu osobinu:

```
PROC Main
  X := 123
  ? "X =", X
  PP (X)
  ? X                // "ABC"
RETURN
PROC PP (x)
  x := 123
  ? x                // 123
RETURN
```

Pri pozivu procedure **PP** kreira se kopija promenljive **X** i stavlja se na stek Naredba **X := 123** menja vrednost na steku o ne originai po vrednost promenljive **X** u glavnom programu ostaje najmanjena. Međutim, u procedari se može koristiti kopija promenljive **X** (u samo primeru zauvim slučajno istog naziva kao i promenljiva u glavnom programu) i ona sadrži i izmenjenu vrednost.

Iz prethodnog primera isiasi da parametre funkcija nije moguće koristiti za vraćanje vrednosti u glavni program (tj u proceduru u kojoj je pozvana funkcija) Za vraćanje vrednosti iz funkcije treba koristiti naredbu **RETURN** U *Clipper*-u ova naredba se označava samo kraj funkcije pomoću ova naredba se može iz funkcija vratiti vrednost proizvoljnog tipa Na primer, sledeća funkcija vraća broj koji predstavlja rezultat računanja u glavni program

```
FUNC Test (X, Y, )
  LOCAL
  
RETURN
```

U samo primeru koje bilo neophodno korišćenje promenljive S za čuvanje rezultata. naredbe **RETURN** omogućava da ne bas povratna vrednost navede i bilo koji izraz pa je to potpuno i ste kao i

```
RETURN "
```

Vrednost koja se vraća može biti proizvoljnog tipa i ne mora biti unapred poznata. To znači da jedna ista funkcija može vraćati kao rezultat u jednom slučaju broj a u drugom string ili datum pa potreb: Potrebu za ovakvim funkcijama je vrlo ta umarena u programima ili svakako treba izbegavati jer otežavaju testiranje programa. Ipak ponekad ovakve svojstve može bita korisno. Sledeći primer je upravo takav slučaj kada se o stoku na koji se mogu upisivati vrednosti proizvoljnog tipa.

```
STATIC aStack := {}
PROC Main
  Push (123)
  Push ("ABC")
  Push (Date())
  ? Pop() Pop(), Pop()
RETURN
FUNC Push (xVal)
  AAdd (aStack, xVal)
RETURN
FUNC Pop
  LOCAL xRet
  xRet := ATail (aStack)
  ASize (aStack, Len( ) - 1)
RETURN xRet
```

Vrednosti koje se prosleđuju funkcijama (kao i rezultati funkcija) zauzimaju slinčkva ograničenja. to mogu biti vrednosti bilo kog tipa, uključujući kodna blokove i nizova. Pri prenosu nizova treba imati u vidu kako *Clipper* radi se stavama, a ima je već bila reči Kada se kao parametar funkcije prosleduje niz, na stek se o stvari stavlja adresa niza Često se prenos elemenja treba vrednosti, pevo se prvi kopija originalne vrednosti (adresa) niza i to se stavlja na stek tako da funkcija prituje originalnim elementima niza, ali prako **kopije niza**. Ova ima dva posledice: 1) ako funkcija promeni neki element niza, promena će se odraziti u glavnom programu 2) ako funkcija promeni vrednost niz – promenljive, originalni na ostale neizmenjen

```
PROC Main
  LOCAL aTest := {1,2,3}
  PP (aTest)          // aTest = {1,2,3}
  ? aTest             // aTest = {1,"A",3}
RETURN
PROC PP (aX)
  aX := Petar()
RETURN
PROC Petar (aX)
  aX[2] := "A"
RETURN
```

Često je potrebno da procedura ipak menja svoje parametre najčešće kod velikih stringova ili memo polja. Tada se prenosi mesto pozicij po adresi. u tom slučaju se ua prenos vrednosti već adresna parametra, pa se sve izmene obavljaju na originalnim vrednostima. Pranos parametara po adresi se označava simbolom „@" (majmun) pre imena promenljive Na primer ako prethodni primer napišemo ovako:

```
PROC Main
  X := "
  PP (@X)
  ? X
RETURN
PROC PP (String)
  String := "EFG"
RETURN
```

u procedura **PP** bila prenata adresa stringa **X**, pa da naredba dodeljivanje dodelili novi vrednost Tako će biti odštampana vrednost **"EFG"**

Tipovi i broj argumenata pri pozivanju funkcija i prilikom deklarisanja ne moraju se slagati za programeru je da ovu osobinu izbegava ili faksirati! Ako se desi greškom, problema može biti veoma neprijatan pogrešna vrednost može rezultovati padom programa ili (još gore) lažnim rezultatima. Opet, ova osobina se može koristiti upotrebiti pr pisanju funkcija u kojima nije unapred poznat broj argumenata ili se ne mora u vak navoditi svaki argument Na primer može se napisati funkcija koja traži maksimum od nekoliko brojeva i poništi se potpuna razvoprovrše sa proizvoljnim brojem argumenata. **Max(1,33) Max(23, 11,33)** itd. Pri pisanju svakvih funkcija koristi se funkcije **PCount()** koja vraća broj stvarno prosleđenih parametara.

```
FUNC Test
  
RETURN
```

Sve ubsira što je u procedari Test deklarisane ari parametra, ses procedura se može pozivati se dva, jedinim ili nijednim pa rametrom na primer **Test(1,2,3) Test(1,2) Test()** Funkcu je **PCount()** vraća broj stvarno zadatih parametara.

VRSTE PROCEDURA I FUNKCIJA

Funkcija koja je deklarisana u jednoj **PRG** datoteci može se pozivati iz svih ostalih funkcija u programu. *Sistem* ne u pravljanje dinamičkim svetiljkama vodi računa o tome koji deo programa (overlay) treba učitati. Međutim, ponekad ovi sasoi na nije zgodna česta je potrebna napravili paku lokalnu funkciju koja je potrebna samo u tekućem modulu (skupu funkcija) Na primer funkcija za brisanje skupa redaka može se zvati **Brisi()** Iste se može nazvati i funkcija za brisanje skupa redaka iz neke druge datoteke – ali, u stvari, to druga funkcija bi morala da nah nazvina različitu stvar Redanja je da se ove dve funkcije različito imenuju, na primer **BrisiRazlika()** i **BrisiStatak()**. Međutim ponekad je ma kako to jednostavna izgledala teška sveiljata nova imena za slične operacije Problem nastaje i u tom slučaju možda dota saboravio nazav funkcija napisane pre dosta vremena i ponovo davaaja istog naziva postaci se sobom ovak neprijatan proiaz kroz ušćun **edit–compile–link**.

Zato *Clipper* poznaje deklaraciju **STATIC** ispred funkcija i procedura tako deklarisane funkcije rade na potpuno isti način kao i obične a tim što su vidljive jedino unutar **PRG** datoteke u kojoj su definisane. **STATIC** funkcija treba koristiti uvek kada je to moguće tima aktivacta logičke celine i onlobađanje neke zamorećaj pitanja. „da li nema već operativno takav naziv?"

EXIT i **INIT** funkcije ahaše se automatsku inicijalizaciju i „gašenje" pojedinih sistema. Naima, ovake deklarisane funkcije se nikad ne pozivaju direktno nego njihovim aktiviranjem automatsku upravlja *Clipper*-ov *run-time* sistem Sva **INIT** procedure koje deklarišete u programu biće izvršene pre svih izvršnih naredbi u programu a sva **EXIT** procedure na samoto kraju programa. Ova je jako zgodna za realizivana sistema koji uobičajeno znaju svoja inicijalizaciju zatim se rad neka operacija i na kraju je obično potrebno „gašenje" zatvaranje otvorenih datoteka i sl. Standardno rešenje je logičan sekvenca tipa pripadaju jednom takvom sistemu cepalj napisanje ui iri dela. prvi deo (inicijalizacija) se poziva na početku glavnog programa, a setvaranja sistema se obavlja na kraju programa. Sada je dovoljno samo ispred procedure za inicijalizaciju dodati **INIT** a sve se može pisati u okviru jedne **PRG** datoteke. Eva jednog primera.

```
PROC Main
  MenuU()
RETURN
INIT PROC MenuI
  ? "MENU IPIT"
RETURN
EXIT PROC MenuExi
  ? "MENU EXIT"
RETURN
PROC MenuU
  ? "MENU"
RETURN
INIT PROC Default
  ? "DNF INIT"
RETURN
```

Startovanje svog programa daje resultat.

```
DNF INIT
MENU INIT
MENU
MENU EXIT
```

Primetite da redosled aktiviranja procedura ne inicijalizaciju ne odgovara redosledu u **PRG** datoteci. Zato svakve procedure treba pisati tako da na savase od redosleda izvršavanja ostalih procedura.

RAZVOJ CLIPPER PROGRAMA

Kada se ponudu nekog standardnog odzora teksta najđe *Clipper* program potrebno je *prevesti*, o setim i *linkovati* dobi jene datoteke u **EXE** datoteku koje se zatim može izvršavati

Prevodilac raspoznaje određene programske konstrukcije i na osnovu njih generiše kod. *Clipper* prevodilac sadrži i *preprocesor* što je svojevrsni programski jezik koji omogućava manipu

lazi se izvornim tekstom programa. Time se povećava mač programskog jezika i programer ima na raspolaganju vaši i zbor naredbi. Osim ovoga pretprocesor može biti veoma korisna alatka za nadgradnju jezika, mogu se dodavati ove naredbe i sl.

Clipper pretvaračke ime zvano opcije pomoću kojih se može utjecati na proces prevođenja programa. U ovom poglavlju bit će tablica reči o tim opcijama i načinu njihovog korištenja.

Vibe **OBJ** datoteke se povezuju u **EXE** korištenjem programa koji se naziva *linker*. Linker koji se upotrebljava uz *Clipper* ima i dodatne mogućnosti, kreiranje dinamičkih overlaja, **PLL** biblioteka, itd.

CLIPPER PREVODILAC

Da bi se program u **PRG** datoteci preveo u **OBJ** koristi se *Clipper* prevodilac. Za te se koristi komanda

```
CLIPPER PROG.PRG
```

Kao rezultat se dobija datoteka **PROG.OBJ** koja je u stvari prevod programa koji se nalazi u datoteci **PROG.PRG**

Ponekad je zgodno više **OBJ** datoteka objedinuti u jednu datoteku. To se može uraditi na dva načina. Prvi je da se odgovarajuće procedure smjeste u jednu umjesto u više **PRG** datoteka, a drugi je da se koristi tzv **skript**. To je standardna ASCII datoteka u kojoj treba staviti nazive svih **PRG** datoteka koje u lane u rezultirajuću **OBJ** datoteku. No prvome nama je u datoteci **SCRIPT CLP** sledeći tekst

```
MAIN
SYSTEM
MENU
HELP
UTIL
```

Kada se izvrši naredba **CLIPPER @SCRIPT.CLP** prevodilac će redom prevesti sve navedene datoteke i sformirati jednu **OBJ** datoteku sa imenom **SCRIPT.OBJ**

Na prevođenje se može utjecati navođenjem raznih opcija pri prevođenju. Na primer ako je potrebna **OBJ** datoteku u *ljudski* informacije potrebne se dibager program treba prevesti sa opcijom */b*:

```
CLIPPER PROG /B
```

Opcije se uvek mogu navoditi na kraju, poslo svih argumenata.

Evo nekih važnijih opcija,

/N Ovo opcija onemogućava da se treba automatski dodeljivati nazivi procedura u **PRG** datoteci; ranije je bilo uobičajeno da prva datoteka sadrži procedure bez naglavlja i takva procedura je automatski dobijala naziv isti kao i naziv tekuće datoteke. Ovu osobinu ne treba koristiti u mnoge je bolje i preglednije ove procedure eksplicitno definisati. Zato ovu opciju obavezno koristite pri prevođenju.

/M Prevodi se samo navedena datoteka. Ako prevodilac pri prevođenju naiđe na naredbe **DO SET PROCEDURE** ili **SET FORMAT** i ako nije navedena ova opcija, automatski se prevodi i ta datoteka. Ova ove naredbe se tretiraju kao zastarele i nije preporučljivo da ih koristite. Ipak, ako ih koristite, navedite opciju /m, kako bi prevodilac prevod samo navedenu datoteku.

/P Prevodilac generiše datoteku sa ekstenzijom **PPO** rezultat rada preprocesora. Ova opciju koristite ako pravite nekoj složenju direktivu preprocesora i želite da vidite šta se dobije kao rezultat.

/Q Pri prevođenju programa, *Clipper* ispisuje listu programa koji se prevodi. Ako se navede ova opcija, lista se ne ispisuju

/B *Clipper* prevodilac na generiše **OBJ** datoteku, već samo proverava da li je program ostalome upravan.

/W Ako se otkrije nedeklarisana promenljiva prijavljuje se odgovarajuće upozorenje. Ovo je veoma korisna opcija koja vam omogućuje da predupredite neke kasnije greške

/A U **OBJ** datoteku se uključuju informacije potrebne dibageru za rad.

/V Svaka nedeklarisana promenljiva se sistematski pridružuje sa **PRIVATE** ili **PUBLIC** promenljivu

Neke od ovih opcije date rečenice koristili Zate je omogu lane da bolje koridene opcije pregledate sa stalno, po kasnije možete dodaveti samo imene. Da bi se to postigle potrebne je izmeniti *DOS* promenljivu **CLIPPERC MD** Na primer

```
SET CLIPPERCMD= /w/n/b
```

Posle ovega će komanda **CLIPPER PROG** biti ekvivalentna se **CLIPPER PROG /w/n/b**

LINKER

Linker služi za povezivanje više **OBJ** datoteka u jednu **EXE** datoteku. Osim ovog standardnog zadatka, *Clipper*-ov linker ima i niz dodatnih mogućnosti Uz *Clipper* se isporučuje linker *RTLink* koji ahiki za povezivanje više **OBJ** datoteka u jednu izvršnu **EXE** datoteku. *RTLink* se može pozvati na dva načina. Kada se koristi prvi način parametri se zadaju u tzv **ahibodnom formatu** – znadenje parametra se određuje na osnovu kjuina reči upred. No primer ima **FILE** ahedi opcijom imena datoteka koje treba linkovati Evo kako se pozivi *RTLink* kada treba povezati datoteke **MAIN OBJ ME NU OBJ** i **SYSTEM.OBJ** u program **APP EXE** i pri tome treba koristiti opciju **VERBOSE**

```
RTLINK FILE main,menu,system
       OUTPUT app
       LIBRARY clipper
       VERBOSE
```

Ako se komanda zadaje iz komandne linije, sve treba pisati u nastavku Drugi način zadavanja parametara je tzv **pozicioni** parametri se razlivajgu zarezom–a znadenje parametara se određuje na osnovu pozicije Na primer prethodna komanda da se može pisati i ovo.

```
RTLINK /VERBOSE FILE main,system,menu,clipper
```

Efekat je potpuna isti i način koji date koristiti je stvar vašeg ličnog ukusa

Komanda za linkovanje se može znatno skratiti Prvo nije potrebna navoditi kako se zovi rezultujuća **EXE** datoteka. Ako se na navede opcija **OUTPUT** podrazumeva se da se **EXE** datoteka zove kao i prva **OBJ** datoteka navedena u spisku Te znači da ako napišemo

```
RTLINK FILE main,menu,system LIBRARY clipper
```

rezultujuća **EXE** datoteka imaće naziv **MAIN.EXE**. Naziv bibioteka se također un mogu navoditi: Develjne ju umesto *DOS* promenljivu **LIB** za spisak kataloga u kojima se nalaze bibioteke i *RTLink* će automatski tražiti potrebne datoteke Najbolje je ova komanda izvršiti u **AUTOEXEC.BAT** datoteci:

```
SET LIB=C:\CLIPPER5\LIB
```

Posle ovega *RTLink* se poziva komandom

```
RTLINK FILE SYSTEM,MENU,UTIL
```

U realnim slučajevima program se sastoji od više manjih modula, taka da ovaj spisak može biti znadgne velike Kunsti uvek upotrebe prethodno komandu za 10-20 datoteka nema mnogo smisla, o i *DOS* ograničava maksimalnu dužinu komande Tada se ačine kao kod prevođenja koristi tzv **skript** datoteka. Sav tekst koji se normalno navodi kao argument *R TLink* e se upisuje n neku standardnu datoteku o linker se poziva sa.

```
RTLINK @FILE.LNK
```

Datoteka **FILE.LNK** sadrži sledeći tekst

```
FILE app,main,system
```

Predmet korišćenja ovih skript datoteka je to što nam nema ograničenje od jedna linije – ovaak datoteka, bibliteka i opcija se slobodno mogu raspoređivati po svoj datoteci

RTLink dižina kao i prevodilac podržava razna opcija pomoć kojih se mogu postići razni efekti pri prevođenju. Spisak svih opcije možete videti u originalnoj dokumentaciji

Linkovanje *Clipper* programa može biti veoma fuge: u slučaju vaših programa, to može trajati i do 2 3 minuta. Ta vrema traje tulko zbog toga što *Clipper* u sebi sadrži i veliki broj sistemskih funkcija se pristupa kazi podataka i razvoji fun kcija za kreiranje korisničkih interfejsa (dodaje i algoritam je kome rade *RTLink* nije previše kvalitetan što se lako vidi poređenjem se nekim drugim linkerima). Sistemske *Clipper* funkcije sa smještene u bibliteke **CLIPPER.LIB**, **EX TEND.LIB** **TERMINAL.LIB**, itd i ulaze n svaki program koji radi se skraćeno i bazom podataka znači, n skaro svaki program. Pošto se linkovanje tih funkcija svak penavlja, na o vaj proces se uvek izneva teški vrema

De bi se ovo izbeglo *Clipper* omogućava kreivanje tzv **pre-link** biblioteka. Osnovna ideja je da se funkcije koje ulaze u više programa unesu u jednu specijalnu datoteku koja se za vrema izvršavanja programa ujlavo po potrebi se ščita. Time se skraćuje vreme linkovanja programa i zmanjuje se veličina **EXE** datoteke Naravno, takvo **FXF** ješteka ne može da radi bez odgovarajuće prošnik (**PLL**) ljiši deke.

Prilikom instalacije *Clipper* kreira PLL biblioteku **BASE52.PLL** koja sadrži osnovne sistemske funkcije iz bibliote ka **CLIPPER.LIB** i **EXTEND LIB**. Da bi se koristila ova **PLL** bibliteka prethodno je potrebni sa *DOS* promenljivu **PLL** še sadrži naziv kataloga u kome se nalazi **BA SE52.PLL**. Na primer ako se **BASE50.PLL** nalazi u katalogu C:\CLIPPER5, treba izvršiti komandu

```
SET PLL=C:\CLIPPER5
```

Onda se *RTLink* poziva komandom

```
RTLINK FILE main,menu,system LIBRARY clipper
```

Izvršavanje programa će biti malo sporije zbog učitavanje podatake **EASE52.PLL**, ali linkovanje će biti znatno kraće Ovo opcija je odlična za testiranje programa. linkovanje se u brzava, a više programa na ščaku koristi jednu zajedničku **PLL** bibliteku, pa se ščuvi i na prostoru

BASE52 PLL sadrik samo sistemske funkcije *Clipper*-a. Obično svaka programer ima neku svoju bibliteku (ili obalan skup) funkcija koju ulaze u svaki program Nema nikakvih prepreka da vi sami kreirate svoju **PLL** bibliteku, u kojo će ući osim *Clipper*-ovih sistemskih i vaše funkcije Za kreiranje takve bibliteke se koristi *RTLink* na sledeći način.

```
PLL.LNK
OUTPUT MySys
LIB clipper  nstend  terminal  dbfntx
RENAME _VDBF,_NDBF,_VDBFNTX,_VDBFCDX,_VMACRO
RENAME _VTERM,_VPICT,_VSETSYS,_VDOS
FILE menu,system,help,util
```

Prethodni tekst treba opisati u skript datoteku (na primer **MYSYS.LNK**) a zatim pozvati *RTLink* se

```
RTLINK @MYSYS.LNK
```

Rezultat neda će biti **PLL** bibliteka **MYSYS.PLL** koja će u sebi sadržati osim standardnih sistemskih *Clipper* funkcije i vaše funkcije iz **OBJ** datoteka **MENU**, **SYSTEM**, **HELP** i **UTIL**. Opcija **REFER** služi se uključivanje funkcija koja se na pozivaju eksplicitno ij navedenih modula, o kazuje de sigurno biti pozvane. Tu su *Clipper*-ova sistemska procedure i morajo biti navedene. Na kraju kreiranja PLL bibliteke *RTLink* javlja nama „nedefinisana" simbola, ka treba ignorisati te procedure de sadržati pravoi vaših programa.

PRETPROCESOR

Neki prevodioci imaju mogudnost raznih operacija se izvornim tekstom programa pre samog prevođenja. Time se u sušini dobija proširenje osnovnog jezika. Deo prevodilaca koji obavlja takve operacije naziva se pretprocesor Pretprocesor koji je ugrađen u *Clipper* osim standardnih operacija omogudava i definisanje novih naredbi Online standardnih mogudnosti (definisanje konstanti, uključivanje kodne datoteka) *Clipper* pretprocesor omogudije jednostavno definisanje novih naredbi, potpuno ravnopravnih se ugrađenim. Mnoge standardne naredbe se realizovane kao preprocesorske direktive

Pretprocesor je veoma sličan pravom programskom jezi ku: ima svoje naredbe, funkcije i promenljive. On se ne bi stva rala nezavisno, naredbe preprocesora se nazivaju **direktive**. Promenljive se obično nazivaju **pseudokonstante**, a funkcije **pseudofunkcije**

Naredbe preprocesora (direktive) se mušnju se normal nim naredbama *Clipper*-a. Da bi se razlikovale, direktive počinju znakom **#**. Najjednostavnija direktiva omogudava definisanje proste zamene teksta u programu.

```
#define K_ESC 27
```

Ova direktiva definiše tekst „27" kao zamenu za pseudokonstantu **K_ESC**. Svuda gde se u programa nađe identifikator **K_ESC** bide zamenjen konstantom 27 Ovo se obavlja prili kom prevođenja programa, pa te nikako nije ista kao kad se definiše globalna promenljiva i koristi je kasnije

Ovakve definicije je veoma preporučljivo koristiti Sve konstante koje koristite u programu (razne kodove tastera, maksimalna vrednost koordinata, itd.) definišite na početku datoteke ovom direktivom. Time neki štedite trud i ako vam ka snije zatreba vrednost neke od ovih konstanti možete morati da „gurnja" svuda po programu, dovoljno je izmeniti samo dedi nicju Uvedenjem konstanta povećavate i čitljivost programa. **K_ESC** je uvek jednjo od broja 27!

Zamena teksta se obavlja pe strogim uslovima: prve se u ulaznoj linji programa ičitavajui identifikatora Ako postoji neki broj odgovara konstanti definisanoj pomoću **#define** vrši se jednostavna zamena. ornosta tekata koji predstavlja ka stanitu pojavite se zamena kaja vrne definisali Pri tome ken stante mora biti iste napisana kao i u definiciji male i velike slova su važnaju u ribur Tako sledeća neće radita

```
#define K_ESC 27

IF taster == K_esc
```

Ova neda pravih neprobrke nepriliu, pe je nam dobro ken niti karvanajlu pa kojoj se ove pseudokonstante pišu velikim slovima.

Tekst za zamenu nije ograničen na jedna broj – kao zamena se može specifičirati ščev string, bas ograničenja.

```
#define USLOV X > 100  AND.  X < 500
...
IF USLOV
```

Da bi se zamena izvršila, konstante mora biti prepoznata kao jedna leksička cela. Take se sa prethodni primer ne može pisati IFUSLOV jer će onda konstrukcija biti prepoznata kao jedan identifikator

S druge strane, preprocesorskaja nije ograničena na identifi katore i izraze: mogude je zamjeti ključne reči operatore pa i ščava naredbe:

```
#define [illegible]  AND  X <= 20
...
ZAKVIRLJAVA
  ? X
...
```

Preiprocesor redom skraćuje svaku linija e zatim je obrađuje

prevodilac. Ako se pri prevođenju navede opcija /P, formirade se datoteka sa ekstenzijom PPO koja sadrži rezultat rada preprocesora. Na primer za prethodni program se dobija:

```
IF A > B .AND. A < 100
  ? A
END
```

Vidi se da je ovde konstanta **ZADOVOLJAVA** zamenjena i naredba *?* funkcijom **QOut**. Menja standardna naredba se ne ovaj način zamenjuju pozivima funkcija. Direktiva koja radi sličn standardne naredbe može naći u **STD.CH** datoteci koja se dobija uz *Clipper*

Definicije mogu biti i višestruke u okviru jedne zamene može se naći drugo ime **DEMO** preprocesor uvek posle obrade ponovo pretražuje ulaz, vrši zamenu i eva teka dok preostali predefinisane identifikatore u naredbu. Na primer:

```
#define K_ESC 27
#define USLOV Ch == K_ESC

IF USLOV
```

Rezultujuća naredba biće **IF Ch == 27**

Direktiva **#define** se najčešće koristi za definisanje raznih konstanti. Uz *Clipper* se isporučuju datoteke sa raznim često korišćenim definicijama. Verovatno najinteresantnije je datoteka **INKEY.CH** koja sadrži definicije kodova tastera. U toj datoteci je kod tastera **<Esc>** definisan kao u prethodnom primeru (K_ESC) kod tastera **<Alt-F1>** ime **K_ALT_F1** i sl

Uz *Clipper* se isporučuju sledeće datoteke sa definicijama.

ACHOICE.CH	– Definicije za funkciju AChoice
BOX.CH	– Definicije raznih okvira za crtanje
DBEDIT.CH	– Definicije za funkciju DbEdit
DBSTRUCT.CH	– Definicije strukture DBF datoteke
DIRECTRY.CH	– Ime za funkcijom Directory
ERROR.CH	– Kodovi grešaka
FILEIO.CH	– Pristup datotekama na nižem nivou
INKEY.CH	– Kodovi tastera
SET.CH	– Razna parametri sistema
SETCURS.CH	– Oblici kursora
MEMOEDIT.CH	– Kodovi MemoEdit funkcije
STD.CH	– Definicije standardnih komandi

Jednom definisane konstante možete koristiti u više raznih **PRG** datoteka. Da ne biste stalno na početku prepisivali definicije možete ukoristiti direktivu **#include**. Funkcija nije se pri tome datoteka „uključuje" u tekstu bal kao da je ona sadržaj prekuzan umesto direktive **#include**. Obično se **#include** direktive postavljaju na početak programa, mada uopšteje nikakve ograničenje i da se postavi na neke druge mesta.

Datoteka navedena u direktivi **#include** mora biti u tekućem katalogu ili se mora navesti puna ime datoteke. Ukoliko sve slične datoteke držite u jednom katalogu možete postaviti *DOS* promenljivu **INCLUDE** teka će sadržaj tog kataloga, ubuduće će direktiva **#include** tražiti datoteke u tom katalogu, ako ih tu nađe u tekućem

Konstante koje su definisite direktivom **#define** vidi u delovi u kojoj je definisane od linije u kojoj je definisana do kraja datoteke, ili prvog definisanje iste konstante ili do poništavanja vrednosti. Vrednost se poništava direktivom **#undef**

```
#undef K_ESC
```

Od sve direktivu korišćenja konstante **K_ESC** generiše grešku.

Preprocesor sadrži i jednostavnu **IF** naredbu pomoću koje možete da utičete na prevođenje programa. Na primer u demonstracionim programima se često jede

```
#ifdef DEMO
  MaxRecord := 100
#else
  MaxRecord := 1000000
#endif
```

Ako se pre svo konstrukcije definiše pseudokonstanta **DEMO** prevodilac će prevoditi naredbu **MaxRecord = 100** a u suprotnom **MaxRecord = 1000000** Zash samo jedna od ova dva naredbe ulazi u izvršni program Naravno umesto jedna naredba može se pisati i čitav blok proizvoljnih naredbi Ova direktive se obično koriste kada je potrebno prilagođavati tekst programa raznim situacijama. Tada naje potrebna pravi ti je jednu varijaju programa za svaku situaciju nego se direktivama **#if** kontrolišu sva krutične mesta. Kasnije je dovoljno definisati ra poželnu programa odgovarajuću pseudokonstan tu (u našem primeru **DEMO**) i ose program će biti usklađen se trenutnim potrebama

Postoji i direktiva **#ifndef** koja provodi naredni blok sa mo ako je nađena konstanta nedefinisana. Pri vrsma nje bilma vrednost pseudokonstante ona može ali na nuva imati vred sosa Na primer

```
#define DEMO

#ifndef DEMO
```

Posle ove direktiva pseudokonstante **DEMO** postaje definisa na, ali se dobija nikakvu vrednost

Definisanje pseudokonstanti se ne mora obavitii samo di rektivom **#define**, nego se to uraditi i pri pozivu prevodioca. Tada je potrebna navesti opciju /D. Na primer

```
clipper test /DDEMO /D
```

Pseudokonstanta **DEMO** dobija vrednost **DA**, potpuna isto kao da je upotrebljena **#define** direktiva. Ovaj način ima i prednosti jer ne morate menjati tekst programa da bi promenili vrednost pseudokonstate.

Direktiva **#define** ima još jednu važnu mogućnost. Deli nisanje pseudofunkcija. Oblika **#define** direktiva vrši prostu zamenu teksta, dok ovaj oblik mnogućuje uvođenje parametra. Ne primer želimo da skramne pisanje oko ispitivanje da li je neki broj u zadatom intervalu. To se može uraditi na sledeći način

```
#define INSIDE(X,Y,Z) (X >= Y .AND. X <= Z)
```

kada se u programu može pisati

```
IF INSIDE( First, 1, 100)
```

Pretprocesor sada opet vrši zamenu teksta, a tim što prethodno prepoznaje parametre **X A ; B** (zamenjuje ih redom tekstom **X, First, Last**) Posle preiprocesiranja se dobija sledeći rezultat.

```
IF (First >= 1 .AND. First <= 100)
```

Pri definisanju pseudofunkcije prve zagrade (otvorena zagrade iza naziva funkcije) nesmo se pisati bez razmaka, jer bi u suprotnom preprocesor to shvatio kao tekst koji treba dodeliti pseudokonstanti *IN*

Među vam se čini da su zagrade koje ograničavaju izraz u prethodnom primeru suvišne Ovde zaista nisu potrebne, međutim može se desiti da prioritet bude narušen ako bi se na jedno strane pozivu pseudofunkcije našla logička operacija veleg prioriteta od **.AND.**, na primer **.NOT.** Zato je uvek bolje stavit zagrade ne treba saboravitii da pseudofunkcija već samo zamenu teksta i do zema načina do rasume sadržaj.

UVOĐENJE NOVIH NAREDBI

Svakako najinteresantnije direktive preiprocesora su **#command** i **#translate**, koje omogućavaju da sa krajnje jednostavni i dosledan način dodajete svoje naredbe u *Clipper*. U suštini rad svih direktiva je gotu sličan direktivi **#define** one na osnovu definisanih pravila vrše zamenu teksta u programima i tako naredbe koje definiše programer prevode u nake već postojeće naredbe ili poerve funkcija. Preiprocesor zatim sve ostalo ovih anališe ovih ključnih prezimena na sebe. Ne primer sledeće direktive dodaju *Clipper*-u **REPEAT** naredbu.

```
#command REPEAT => WHILE .T.
#command UNTIL <X> => IF !(<X>) ; EXIT ; END ; END
```

Posle svakvih direktiva, u programu se može naći mešovite konstrukcija, potpuno ravnopravne sa **WHILE** ili **FOR..NEXT** naredbama.

```
REPEAT
  ? A
UNTIL A >= 100
```

Preiprocesor prethodnu konstrukciju prevodi u beskonačni ciklus koji se prekida naredbom **EXIT** kada uslov bude zadovoljen. Korišnik nema nikakvih komplikacija oko preprocesovanja ključnih reči, izraza, itd – preiprocesor sve to automatskim radi.

U radu preiprocesora je svakako najveći problem prepoznavanje izraza, ključne reči je lako „uparšti" ali *Clipper* nezi nogu biti vrsma kompliksvani Konstrukcija **<IDend>** (izvo za marker) u definiciji zamenjuje proizvoljan izraz Na primer mala rečs je on zamenjen stringom „i<= 100" Ove preskivanje je najprostija – izraz koji se pojavi u ulazu treba prekinuti na nekv. Međutim za definisanje vakih naredbi potrebno je izvršiti i neku transformaciju ulaza. Zato direktiva **#command** preiprocesije razne oblike ulaznih i izlaznih markera.

Na ulazi (na leve strane definicije) mogu se pojaviti sledeći markeri

<!D>	Ova je najjednostavniji mariner i posu se regularni Označava jedan verar u ulaznom tekstu. Na primer ako se napiše **<Exp>** to može zamenjuje izraz **X < 12 .AND. X > 0**
<!D, ...>	Marker liste. Koristi se kad treba u ulazu „prepoznati" niz izraza razdvojenih zarezima. Na primer, marker **<List, ...>** prepoznaje sledeći tekst. **x, x+y "ABC", x < 3.**
<!dx,y,z>	Marker skupa rezervisanih reči. Koristi se kad je potrebno označiti tekst između nekoliko rezervisanih reči, na primer **ALL, REST** i sl.
<*!D*>	Zamenjuje kompletan tekst do kraja naredbe, bez obzira na sustalnosi sadržaj
<(!D)>	Marker „preiiresog" izraza. Koristi se kod naredbi kaa što je **USE** i sl.

Kada preiprocesor obrađuje ulazni tekst, definisiju navedena na leve strani oznake => upuruje se na ulazni tekstom. Ako to pervaaja se tapo, prevodilac javlja grešku.

Tekst koji „preparsan" marker navedan na leve strane transformiše se u rezultujuć izraz. Kakav će biti rezultat određuje odgovarajuć marker na desne strane. Za desnu stranu definicije mogu se koristiti sledeći markeri

<!D>	Regularni izlazna marker Ulazni tekst se bez mnsma preslikava u izlaz
#<!D>	Frost string izlazni marker Ulazni tekst se prevaza u string dodavajom oznaka " na početak i kraj
<"!D">	Normalni string izlazni marker Ulazni tekst se pretvara u string. Ako je ulazni tekst nije izraza, svako pojedinačno se pretvara u string.
<(!D)>	Pametan string izlazni marker Radi isto kao normalni string izlazni marker osim što podešava *Clipper*-ove preiprocesorske izraze
<{!D}>	Blok izlazni marker Ulazni tekst se pretvara u kodni blok. Na primer ako je na ulazu tekst **X + 3** rezultat će biti **{ || X + 3 }**
<.!D.>	Logički izlazni marker Ako postoji ulazni tekst, prosliiava se u logičku vrednost **T.** a u suprotnom u **.F.**.

Ako često koristite neku funkciju sa više argumenata, dobra je definisati odgovarajuću naredbu direktivom **#command** i tako zaokružiti jednu celinu. To će vas podsetiti i na to da uvek pogađate da je, na primer, neki parametar u poziv funkcije početni broj opcije i sl – kljućna reč u naredbi se pamte mnogo efikasnije, a doseviinas se i ponekad oslobađate od redoslede koda za u pitanju neobaveznu parametri.

PROGRAMIRANJE KORISNIČKOG INTERFEJSA

Osim standardnih naredbi i funkcija, *Clipper* sadrži i specijalne naredbe za kontrolu ekrana, tastatura i baze podataka. U ovom delu će biti govora o naredbama za kontrolu tastature i ekrana. Ove dve grupe naredbi su vesma srodnim jer se pomoću njih odvija kompletna komunikacija sa korisnikom. U nas podataka, ispisivanje na ekranu ili štampaču i sl Oči kvalitete ove komunikacije ogkavnom zavisi i kvalitet i lepo programa – poslovni programi kakvi se obično pišu na *Clipper*-u spoju vesma ostanovivni komunikaciju sa korisnikom.

ISPISIVANJE TEKSTA

Kada je potrebno ispisati neki tekst na ekranu, koriste se naredbe *?* i *??* Ove dve naredbe služe za ispisivanje proizvoljnih vrednosti na ekranu Na primer

```
? "ABCD"
```

ispisuje string "ABCD" na ekranu. Može se zadati i niz izraza.

```
? "ABCD", DATE(), PCount(), 3.1415
```

Pri tome se zadati izrazi razdvajaju ispisuju, jedan za drugim. Između svake dve vrednosti ima ostavljena jedna praznina.

Ispisivanje počinje od *tekuće pozicije kursora* – kada se tekst ispisuje, kursor se pomera i sledeći ispis počinje od mesta gde je kursor poslednji put „ostavljen" Ako kursor nije na početku reda naredba *?* prvo pomera kursor na početak narednog reda, a zatim ispisuje zadate izraze Kursor ostaje na poslednjoj ispisanoj znakova u redu. Na primer

```
CLS
? "ABC"
```

Tekst **"ABC"** biće ispisan u prvom redu (naredba **CLS** pomera kursor na početak prvog reda) a tekst **"DEF"** u drugom redu Ako navedete naredbu *?* bez argumenata, kursor će preći u sledeći red Kada se naredba *?* izvrši a kursor je u poslednjem redu. ekran se skroluje za jedan red na gore Na taj način se ispisuju sem kadbene podataka.

Često je potrebno tekst ispisivati od tekuće pozicije kursora, a ne od početka reda? Za to se može koristiti naredba *??* koje radi kao *?* osim što pre ispisivanja na pomera kursor a sledeći red Na primer

```
CLS
? "ABC"
?? "DEF"
```

ispisaće na ekranu tekst ABCDEF u jednom redu – posle ispisa stringa **ABC** kursor je ostan iza slednjeg **C** i naredba *??* je tu nastavila ispisivanje.

Naredba *?* i *??* se prilikom prevođenja „protprocesiraju" u

posve funkcija **QOut** i **QQOut**, pa je potpuno svejedno da li pišete **? nVar** ili **QOut (nVar)** Predmet funkcije je što se manje upotrebljavati u izrazima i kodnim blokovima a šanse da kasnije bilo što

FORMATIRANJE IZRAZA

Kao što smo već vidjeli funkcije **QOut** i naredba **SAY** mogu da ispišu vrednosti različitih tipova podataka. Bez obzira na to često je najjednostavnija upravo stringovna – postoji veliki broj funkcija koje vaoma ukalsavaju formatiranje stringova. Takve su funkcije **Substr**, **Left**, koje iz stringa izdvajaju zadate delove. Postoje još jedna vesma važna funkcija koja omogućava formatiranje podataka proizvoljnog tipa, funkcija **Transform**.

Ova funkcija podatak iz bilo kog tipa pretvara u string, slično kao što funkcija **Str** pretvara brojeve u stringove. Međutim **Transform** ima daleko veće mogućnosti od funkcije **Str**. Prvo, prihvata i ostale tipove podataka, a na zaraz brojeve, drugo, omogućuje daleko preciznuju kontrolu nivga što se dobija kao rezultat.

Transform funkcija se kontrolište na dva načina, preko stringova za formatiranje i funkcije se formatiranje. Stringovi za formatiranje služe za operacije nad simbolima ulazne vrednosti. String za formatiranje je, u stvari, niz simbola od kojih neki imaju specijalno značenje. Na primer, simbol „0“ u formatirajućem stringu označava cifru. Ako se „upotrebi“ na format stringa, i na mestu toga se pojavi izlaz. Na primer

```
? Transform (1234, "0000") // "1234"
```

Prethodna naredba ispisuje rezultat formatiranja broja 12345 formatirajućim stringom **"999,999"**. Svaka ulazna cifra se preslikava u cifru raspoređenu po zadatoj šemi, a mesto nataje tako da se kao rezultat dobija **12,345**. *Clipper* podržava još specijalnih simbola.

A,N,X,9,#	Označava jedan simbol ulaznog podatka. Rezultat nevas od tipa ulazne vrednosti
L	Prikazuje logičku podatak kao T ili F
Y	Prikazuje logički podatak kao Y ili N
!	Konvertuje slova u odgovarajuće velika
$	Prikazuje $ umesto vodećih nula u broju
*****	Prikazuje * umesto vodećih nula u broju
.	Odredjuje decimalnu tačku
,	Zarez

Evo nekih primera upotrebe prethodnih specijalnih simbola.

```
? Transform (1234.56, "9999,999.") // "  1234.56"
? Transform (1234, "$$$$$$") // "  $$$$1234"
? Transform (1234, "*****") // "**1234"
? Transform (-1234, "#######") // "-  1234"
? Transform ("Hendrt", "A# A# A#") // "H  e  n  d  r  t"
```

Kao što se vidi u poslednjem primeru, kada se formatira string, simbol u formatu koji nije specijalan jednostavno „prerkrije“ odgovarajuće ulazne simbole.

Drugi način formatiranja je preko funkcija na formatiranje. Funkcije za formatiranje dobiju na ulazu vrednost, a na posebnoj mestu na simbole. Zadaje se takođe u okviru stringa se formatiranja, samo što se pre kodova funkcija nalazi simbol **@**. No primer da bi se ova slova u ulaznom stringu konvertovala u velika, nadaje se sledeća naredba.

```
? Transform ("Hendrt", "@!") // "HENDRT"
```

Funkcija **Transform** raspoznaje sledeće kodove – osnaka funkcija na formatiranje

S	Broj poravnava u levo
C	Prikazuje CR posle pozitivnog broja
D	Prikazuje datum u tekućem formatu
E	Prikazuje datum u evropskom formatu bad brojeva
R	Simboli bez značenja u stringu se formatiranje na brisi u ulazu, već se ubacuju izmedu
X	Prikazuje DB posle negativnog broja
Z	Prikazuje nule kao blanko simbole
(	Zatvara negativne brojeve u zagrade
)	Zatvara negativne brojeve u zagrada bez vodećih blanko simbola
!	Sva slova konvertuje u velika

Posle oznake za format funkciju (**@**) može se odjednom navesti i više kodova za funkcije. Na primer string "@BC" znači da broj treba poravnati u levo i da na kraju treba ispisati CR ako je broj pozitivan. Ponekad je potrebna kombinovati funkcije i simbola za formatiranje tada treba prve pisati fun kcije, a zatim simbole posle jednog blanko simbola. Na primer

```
Transform (dRodjen, "@L AA-AAA/AAA")
```

Pri tome treba kao u ovom da funkcije na formatiranje imaju prednost nad simbolima.

Svi prethodni mogli su direktno mogu koristiti i u **SAY** naredbi korišćenjem opcija **PICTURE**. Kada je potrebno formatirati ulaz **SAY** naredbe dovoljna je navesti opciju **PICTURE** na odgovarajuće stringom na formatiranja. Na primer

```
@ 2,10 SAY 100 -- PICTURE "999,999"
```

Prethodna naredba je potpuno kao kako kada bi se broj prethodno formatirao pomoću funkcije **Transform** a zatim odštampo pomoću obične **SAY** naredbe

KONTROLA KURSORA

Položaj kursora na ekranu se u *Clipper*-u označava njegovim koordinatama. Gornji levi ugao predstavlja se koordinatama (0,0) U opštem slučaju koordinate su (X,Y), gde je X broj reda (numeracija počinje od 0) a Y broj kolone To znači da na koordinate uglova na vašem ekranu (0 0) (24 0) (24 79) i (0 79) Ove je standardno i najčešće se koristi Međutim na vi deo karticama koje omogućavaju različite tekstualne modove, mogućе je koristiti i neralnuju se vido redova i kolona Na primer ako imate VGA karticu, možete koristiti 43 umesto 24 redova i kolona. Funkcijom **SetMode** se video kartice prebacuje in jednog u drugi mod. Na primer **SetMode (43, 80)** povećava broj redova na 43 umesto standardnih 24. Tada numerisanje redova počinje od 0 a poslednji red je 42 Naravno, sve to ne može da raditi na Hercules-u i na sličnim video karticama koje ne podržavaju povećanje broja redova. Kako bi vaše aplikacije bi ja što opšija, dobro je da vebčinu ekrana određujete za vreme izvršavanja programa, koristeći funkcije **MaxRow** i **MaxCol** koje vraćaju maksimalni broj reda i broj kolona. Na taj način će vam program raditi bez obzira na izabran video mod

Često je potrebno pročitati tekuću poziciju kursora. Za to se koriste funkcije **Row** i **Col** koje vraćaju trenutnu poziciju (red i kolonu) kursora bez obzira da li je kursor stvarno vidljiv Naravno, kada upravljate nad vrednosti na ekranu, mnja se pozicija kursora, pa i vrednosti koje vraćaju funkcije **Row** i **Col**.

Vidljivost i oblik kursora se takođe mogu kontrolisati programski. Funkcija **SetCursor** prihvata on broj između 0 i 4 i na osnovu toga postavlja jedan od pet različitih oblika kursora:

0	isključuje kursor
1	
2	
3	
4	

Ubičajene je da se različiti oblici kursora koriste kad je potrebna osnažiti pralnak na mesta za prekucavanje u insert mod i al.

Naredbe **SET CURSOR ON** i **SET CURSOR OFF** na potpuno ekvivalentna pozivima funkcija **SetCursor(1)** i **SetCursor(0)** U stvari, naredbe **SET CURSOR** prilagovavaju prvredi i odgovarajuće pozive funkcija

ISPIS SA POZICIONIRANJEM

Osim naredbi **? i ??** (odnosna funkcija **QOut** i **QQOut**), *Clipper* ima još dasta naredbi i funkcija koje služe za kontrolu ekrana. Korišćenjem tih funkcija i možete vaše programe oprematiti vesma atraktivnim izgledom

Fenalazi je potrebna pomeriti kursor na novu poziciju, ben ispisivanja podataka Funkcija kojom se to postiže zove se **SetPos**. Na primer **SetPos (9 30)** pomera kursor u četvrti red, trideset prve kolone. Sledeći program jednaje kako radi ove funkcije. Ispisuje slova "A" presecajući ekran

```
PROC Main
  LOCAL i
  FOR i := 1 TO MaxRow ()
     SetPos (i,i)
     ?? "A"
  NEXT
RETURN
```

Posle se najčešće kursor pomare da bi se upisao neki podatak. *Clipper* ima naredbu koja istovremena pomera kursor na novu poziciju i ispisuje vrednost to je **SAY** naredba. Na primer

```
@ 0,79 SAY "ASCII"
```

Prethodna naredbe prve pomera kursor na poziciju (3,10) a zatim ispisuje string „Abeceda“ Naravno, osim stringova mogu se upisivati i drugi tipovi bi logičke vrednosti Naredba **SAY** se preformatira u posive funkcija koje prve pomeraju kursor na zadatu poziciju pa zatim upisuju vrednost

Za brisanje ekrana se koristi naredba **CLEAR TO** koja ima sledeću sin taksi

```
CLEAR [TO nTop, nLeft [TO nBottom, nRight]]
```

Pravougaonik na ekranu se koordinatama uglova (x1,y1) i (x2,y2) bute ukriran posle sve naredbe Naredba **CLEAR TO** je (bar u većini bog ovih naredbi) realizovana preko odgovarajućih funkcija. Funkcija koja može zamenite sve naredbe nasiva se **Scroll** Ona može sem brisanja ekrana, da pomera („skroluje“) deo ekrana na gore dole, leve ili desno Na primer prethodna naredba bi bila ekvivalentna se

```
Scroll (x1,y1,x2,y2, 0)
```

Poslednji argument (0) označava da treba ukriranti zadati pravougaonik. Ako se ovo argument navede pozitivan broj ceo ekran se pomera na gore o negativan na dole za zadati broj redova. Eva kako možete na interesantan način ukriranti ekran.

```
PROC Clear
  LOCAL i
  FOR i := 0 TO 12
     Scroll (i,i,24-i,79-i,1)
     Scroll (i,i,24-i,79-i,-1)
  NEXT
RETURN
```

Ova procedura postepena skroluje sliku na ekranu i to gornju i donju polovinu Gornja polovina se pri svakoj iteraciji pomeri na jedno mesto na gore a donja na dole. Kada se ove slika pomeri 13 puta, ekran će biti potpuno čist. Ako je potrebno ukloniti ceo ekran, treba koristiti naredbe **CLS** koja se takođe preiprocesira u poziv naredbe **Scroll**

Kada planirate izgled ekrana, dobro je predvideti da se neki podaci naglase na izvestan način te je preporučljivo zbog korisnika–jer se na taj način skreće njegova pažnja na najvažnije podatke na ekranu. Jedna od najpopularnijih načina se te je crtanje okvira od simbola iz standardne *ASCII* tabele. Takve okvire na najrate crtati kasnijitujuće kodove u *ASCII* tabeli dovoljno je da upotrebite sledeću naredbu

```
@ x1,y1, x2,y2 DOUBLE
```

Tako će sadati pravougaonik na ekranu biti nertan jednostrukom linijom Ako želite da okvir bude nacrtan dvostrukom linijom dodajte na kraj ključnu reč **DOUBLE**

```
@ x1,y1, x2,y2 DOUBLE
```

Pesle izvršenja naredbe, kursor se pomara u gornji levi ugao nacrtanog pravougaonika, tako da odmah možete početi na i spesavanjem podataka. Ova naredba na manja unutrašnjost pravougaonika.

Pravougaonici (okviri) se mogu crtati i od drugih simbola iz *ASCII* tabele a takođe se može kontrolisati i unutrašnjost nacrtanog pravougaonika. To se ostali naredba **BOX**.

```
@ x1,y1, x2,y2 BOX "SimboliOkvira"
```

Sada se pravougaonik crta korišćenjem zadatih simbola („a“ „b“) pri čemu prvi simbol služi za crtanje gornjeg levog ugla, a ostali se crtanje ostalih elemenata, a onaru kazaljke na satu Poslednji simbol se koristi na popunjavanje celog ekvira. Na primer naredba.

```
@ 0,0, 8,8 BOX "Blanko"
```

ispisuje pravougaonik.
Naredba **BOX** se, kao i ostale slične naredbe, preiprocesira u funkciju **DispBox** koja ima isto dejstvo i mogućnosti kao i opisana naredba.

MENIJI

Sa kazama podataka mogu raditi (a obično i rade) i korisnici koji nisu profesionalni programeri Zato se pri pisanju programa mora predvideti i red se korisnici koji ne poznaju a autrazitnu logiku programa Ne vama je da program bude što je moguće jednostavniji za korišćenje ali da pri tome na izgubi na mogućnostima. Sve u svemu komunikacija se korisnikom je vesma vaina. S obzirom da programi koji se pišu u *Clipper*-u najčešće imaju mnogo ulaznih podataka, uvek postoji pažnja obrazite na dao programe koji služi za komunikaciju se korisnikom. Ovaj deo programa se naziva **korisnički interfejs** (*user interface*) *Clipper* ima modan skup naredbi i funkcija čijim korišćenjem možete napravite veoma kvalitetan korisnički interfejs. Korišćenjem tih funkcija mogu se brze i jednostavno kreirati mnogi za izbor opcija mnoga za interaktivni nasa podataka itd.

Najjednostavniji način da se korisniku omogući izbor jed na od nekoliko zadatih opcija je **scezi** na ekranu se ispisuju mogućnosti a korisnik pomera kursor di neku oznaku na opciju koju želi da izabere. Pogledajmo kako se to radi korišćenjem naredbe **MENU TO**

```
@ 0,0 PROMPT "Kreiranje"
@ 1,0 PROMPT "Brisanje "
@ 2,0 PROMPT "Kraj rada
MENU TO nChoice
```

Prve je potrebno specificirati koja na opcije menija i na kojim koordinatama se ispisuju. To se radi naredbom **PROMPT** @ **x,y, PROMPT** cString na Kuračnom (x,y) ispisuje string **cString** (opcija menija) a naredba **MENU TO** aktivira meni Sva prethodna definisana stavke postaju opcije menija i korisnik može tastaturu za pomeranje kursora da selektuje bilo kojanu opciju. Kada izabere, dovoljne je da pritisne taster **<Enter>** i izbor se završava. promenljiva **nChoice** dobija broj vrednost broj izabrane opcije Ako korisnik pritisne **<Esc>** na vesma izbera, prispostavlja se da korisnik želi da odustane od izbora opcije i promenljiva **nChoice** tada dobija vrednost 0

Svakoj stavki menija može se dodaliti i male duži tekst, u objašnjenje opcije, koji se se vrema izbere ispisuje obično u poslednjem redu ekrana Ne primer može se pisati

```
datoteka"
```

Znaš, kada korisnik kuša pomeriti kursor na opciju **Delete**, u dnu ekrana (ili na mestu koja se definiše naredbom **SET MESSAGE**) će pisati detaljnije objašnjenje te opcije.

Ova naredba se obično koristi u kombinaciji sa **CASE** naredbom na sledeći način

```
PROC Main()
  LOCAL nChoice
  WHILE .T.
    @ 1,0 PROMPT "Unos slога "
    @ 2,0 PROMPT "Izmena slога"
    @ 3,0 PROMPT "Brisanje     "
    MENU TO nChoice
    DO CASE
      CASE nChoice == 0 ; EXIT
      CASE nChoice == 1 ; AdUnesi()
      CASE nChoice == 2 ; ChgSecord()
      CASE nChoice == 3 ; DelSecord()
    END CASE
  END DO
RETURN
```

Prva se pomoću **MENU TO** naredbe korisniku omogućava da izabere jednu od ponuđenih opcija, a zatim se se ostvoni broja izabrane opcije prosleđuje dalja akcija, pozivu se odgovarajuće procedura na dodavanje unosu ili brisanje sloga. Kada korisnik izabere jednu opciju i izvrši se izvršena akcija, ga se kontrola (zahvaljujuće naredbi **WEDLE .T.**) ponovo vraća **MENU TO** naredbi tako da korisnik ponovo može da bira novu akciju. Te se ponavlja sve dok korisnik ne odtraži izlaz iz tog menija. Tada promenljiva **nChoice** dobija vrednost 0...pa se ulazi prekida.

Dobro proučite prethodnu konstrukciju vezana je praktično da vaš program imaju isti ili bar slično strukturu. Tako ćete tako dodavati nove opcije u vaš program, a strukturu ćete dalja ostati prepoznata se izmene.

FUNKCIJA ACHOICE

Menije mogu biti i nanogo složeniji nego što predvida korišćenje naredbe **MENU TO** neka opcije mogu biti trenutno nedostupne opcija može biti više nego što može stati na ekran hod izbora neke opcije treba izvršiti neku posebnu akciju, itd. Zato u *Clipper*-u postoji funkcija **AChoice** koja omogućava izradu veoma fleksibilnih menija.

Najednostavniji način korišćenja **AChoice** funkcije je sledeći

```
[illegible]
```

Argumenti funkcije se koordinate „prozora" u kome treba ispisati opcije menija i niz sa imenima koji predstavljaju opcije menija. Na primer

```
[illegible]
to"}
```

Kada se ekvivira ova naredba, u zadnjem delu ekrana kao i opisano sadaste opcije i korisnik može testerima za pomeranje kursora da bira opciju menija. Kada izabere opciju pritisak se taster **<Enter>** pritisak izvršavanja funkcije i promenljiva **nItem** dobija vrednost rednog broja izabrane opcije Ukoliko korisnik odustane od izbora i pritisnu taster **<Esc>** funkcija vraća vrednost 0 Isti efekat se postiže ako korisnik pritisna taster se onemkom levo ili desno Ukoliko sve opcije ne mogu da stanu u prozor gri izbora se vrši „skrolovanje" u prozoru je prikazan deo opcija, a ostala se pojavljuji pri skrolovanju.

U sledeći tabeli je dat spisak svih tastera i odgovarajućih komandi koja rasponanje funkcija **AChoice**:

<Up>	Prethodna opcija
<Down>	Naredna opcija
<Enter>	Izbor opcije
<Esc>	Odustajanje od izbora
<Left>	Odustajanje od izbora
<Right>	Odustajanje od izbora
<Home>	Prva opcija
<End>	Poslednja opcija
<Ctrl-Home>	Prva vidljiva opcija
<Ctrl-End>	Poslednja vidljiva opcija
<PgUp>	Prethodna „strana" opcija
<PgDn>	Naredna „strana" opcija
<Ctrl-PgUp>	Prva opcija
<Ctrl-PgDn>	Poslednja opcija

Ovim nabrojanim tastera se izbor može se koristiti i prva slova opcije Na primer ako je jedna od opcija menija „Delete" i su jedna opcija pra ova ne počinje slovom „D" pritiskom na taster se slovom „D" kursor se pomera se ovu opciju Naredni pritisak na taster „D" izazvat će skok na sledeće opciju koja počinje na „D"

AChoice omogućava da pojedina opcije privremeno zabranite se izbor Potrebno je kreirati niz logičkih vrednosti (size dužina lista i svaki opcijski koji će sadržati vrednost T se dostupan opcije, a F za one koja se ne mogu izabrati Pri izboru te kursor jednostavna preskakan opcija kod kojih je vrednost F (netačna) Na primer

```
[illegible]
```

U ovom primeru se prva proveruva rezultat funkcije **System()** – ako ona vrati vrednost tačno, korisnik će u sledećem meniju moći da bira opciju „Delete user" jer je odgovarajući element niza **aPriv** vrednost T a u suprotnom neće moći.

Niz **aPriv** se mora sadržati samo logičke vrednosti. u njemu možete upisivati I kompletne izraze koji će se makro operatorom sračunavati kada treba odrediti da li je opcija „dostupna" ili ne. Na primer niz **aPriv** se može zadati i na sledeći način.

```
[illegible]
```

Ukoliko je potrebno zabraniti/omogućiti izbor svih opcija u meniju umesto nam koji sadrži N istih vrednosti može se on dati i samo jedna vrednost. T ili F u zavisnosti od toga da li korisniku treba omogućiti ili zabraniti izbor

Ovim nipo iscrpljene sve mogućnosti funkcije **AChoice**. Njene napredne moć proizilazi iz definisanja niv korisničke funkcije **Kada se desi** ova funkcija, funkcija **AChoice** jej „pruprista" dve kontrole i na taj način se mogu ostvariti veoma korisni i interesantni efekti

Pri pozivu funkcije **AChoice** korisnička funkcija se zadaje kao sedmi parametar i te mora nazir funkcije **ben** parametara. Primetite da je pri pozivu funkcije **AChoice** prosleđen šesti parametar (nje vrednosti se pričuje opisivanja)

Korisnička funkcija se poziva posle svakog tastera koji prihvata korisnik. Ako tastar funkcija propozna kao svoju komandu, prve se ozvršava odgovarajuće akcija, a zatim se posl va funkcija korisnike kaja određuje šta želja treba raditi Ako funkcija ne prepozna komandu odmah se poziva funkcija koja je korisnik zadao Kada **AChoice** funkcija radi se zadatom korisničkom funkcijom neke upućivana komanda se se prepoznaja svi tastari koji služe za kraj rada (**<Esc>** **<Left>**) gube svoje značenje.

Pri pozivu korisničke funkcije automatski se prosleđuju tri parametra.

nMode	Broj koji označava trenutno stanje u kome je funkcija (bira se opcija, pritisnut je nepoznati taster) Na osnovu svog broja se odreduje šta se trenutno dešava na ekranu. Na primer ako je **nMode** = 1 korisnik pokušava da pomeri kursor pre prve opcije u meniju
nCurrent	Broj opcije na kojoj je trenutno kursor
nRow	Broj reda u prozoru u kojem se trenutno nalazi kursor

Da bi se sklalale analiziranja se mogućim vrednostima promenljive **nMode**, ux *Clipper* se isporučuje datoteka **ACHOICE.CH** koja sadrži simbolička konstantne se vrednosti koja „vrača" **AChoice.** Na primer umesto da pišete **nMode == 0** dalske je pregledaje **nMode == AC_IDLE** Na sbo I je je kanan primer korišćenja funkcije **AChoice** se korisničkom funkcijom.

Na osnovu ulaznog moda funkcija **AshFunc** preduzima odgovarajuću akciju. ako je uobičajen prelaz upućuje se parametri funkcija **AshFunc**, ako je korisnik pokušao da kursor pomeri posle kraja menija izlazi se odgovarajuće upozorenje i sl.

Kada se završi analiza stanja, potrebna je funkcija **AChoi**na se neki način zapažiti dalje akcije Ta se radi tako šta funkcija **AshFunc** vrati određenu vrednost. Ova vrednosti su takođe definisane u datoteci **ACHOICE.CH.** Na primer: ako korisnik pokuša da pomeri kursor posle poslednjeg elementa menija, posla upozorenja upozorenja funkcije **AshFunc** vraća vrednost AC CONT čime se funkciji **AChoice** nalaže da ova nalno nastavi rad.

Postoji petnju zaustavje ukoj kada je **nMode** = **AC_EXCEPT** To se dešava kada funkcija **AChoice** se prepoznae pritisnuti taster Na primer korisnik pritisne taster **<F1>** Pošto taj taster nema ništakvo značenje za funkciju **AChoice**, funkciji **AshFunc** biće prosleđeno **nMode** = **AC_EXCEPT** Sada je on korisnikovoj funkciji da otvori šta treba uraditi Te to u datom primeru obavlja tako što se prvo pomoću funkcije **LastKey()** odredi koji je taster poslednji pritisnut, a zatim se pomoću **CASE** naredbe određuju dalja akcija. Primera radi, ako je pritisnut taster **<Enter>** funkcija se nalaže da prekine rad i vrati redni broj izabrane vrednosti (**AC_SELECT**)

Parametar **nMode** može imati sledeće vrednosti (u stvari ovo su simbolične konstante definisane u datoteci **ACHOICE.CH**)

AC_IDLE	Normalan prolaz Javlja se kad AChoice ubrzdi zadatu komandu i zatim ponovo preuzima funkcija.
AC_HITTOP	Korisnik pokušava da pomeri kursor ispa prve opcije u meniju.
AC_HITBOTTOM	Korisnik pokušava da pomeri kursor iza poslednje opcije u meniju
AC_EXCEPT	AChoice nije preponeo pritisnut taster i poziva korisničku funkciju.
AC_NOITEM	Na postoji ni jedna opcija menija kojoj je dozvoljen pristup

Korisnička funkcija može funkciji **AChoice** vratiti sledeće vrednosti.

AC_ABORT	Prekini rad, vrati 0 kao rezultat izbora
AC_SELECT	Prekini rad, vrati redni broj tekuće opcije kao rezultat izbora
AC_CONT	Nastavi normalno rad
AC_GOTO	Pomeri kursor na sledeću opciju koja počinje se slovom koje je upravo otkucano

SLIKA 1: Primer korišćenja funkcije ACHOICE

```
[illegible]
#include "Achoice.ch"

PROC Main
  LOCAL nItem
    LOCAL aItems := {"Append", "Copy", "Delete", "Move",
"Rdme"}

  CLS
  nItem := AChoice (4,58,7,47, aItems,, "AshFunc")
RETURN

FUNC AshFunc (nMode, nCurrent, nRow,
  LOCAL nRet := AC_CONT, ch

  DO CASE

    // Uobičajen prolaz
    CASE nMode == AC_IDLE
      Msg ("MODE: " + Str(nMode,2,0)
           + "   CURR " + Str(nCurrent,2,0)
           + "   ROW " + Str(nRow,2,0))

    // "Usarac" u vrh menija
    CASE nMode == AC_HITTOP
      Msg ("""  Pocetak menija """)

    // "Usarac" u dno menija
    CASE nMode == AC_HITBOTTOM
      Msg ("""  Kraj menija """)

    // Nema ni jedne opcije koja
    CASE nMode == AC_NOITEM
      Msg ("""  Nema opcija za izbor """)
      nRet := AC_ABORT

    // ACHOICE nije prepoznala
    CASE nMode == AC_EXCEPT

      // Koji taster je pritisnut
      Ch := LastKey()

      DO CASE

        // Exit
        CASE Ch == K_ENTER
          nRet := AC_SELECT

        // Quit
        CASE Ch == K_ESC
          nRet := AC_ABORT

        // Ništa sa opcijo
        CASE IsAlpha (Chr(Ch))
          nRow := Ch

        // Komanda
        OTHERWISE
          Qmo ("""  Komanda
      END CASE

  END CASE
RETURN nRet

// Funkcija prosleđuje
PROC Msg (cTxt)
  @ 24 0
  @ 24 0 SAY cTxt
RETURN
```

Funkcija **AChoice** dozvoljava selektovanje još dva parametra (iza naslova horizontalne funkcije) početni broj opcije i poruku ju unutar prozora. Ovi selektovani parametri služe za promenu podešenog položaja kursora kojim se selektuje stavka.

ČUVANJE SADRŽAJA EKRANA

Svaki ispis na ekran neposredno uništava prethodni sadržaj. Vrlo često je potrebno stari sadržaj ipak povratiti. Na primer ako je na ekranu nacrtana neka slika (lego firme ili ai.) startovanje naredbe **MENU TO** omogućava izbor jedna od jedanaest opcije ali se time briše sadržaj dela ekrana. Ovakve situacije se javljaju veoma često pa sete *Clipper* ima funkcije **SaveScreen** i **RestScreen** koje služe za čuvanje i kasniju restauraciju dela ekrana.

Kada je ekran u standardnom tekst modu, svakom simbolu na ekranu odgovaraju dva bajta u tzv. *video* memoriji prvi bajt sadrži ASCII vrednost simbola, a drugi bajt tog simbola (često se ovo naziva *atribut*) Kada se nešto ispiruje na ekran, menja se sadržaj video memorije Funkcija **SaveScreen : RestScreen** određenu segmentu video memorije kopiraju u string promenljive i nkrnuto

Funkcija **SaveScreen** kopira zadati deo ekrana u string promenljivu No primer.

```
cEkran := SaveScreen (0,0,24,79)
```

kopira deo video memorije koji odgovara pravougaoniku (0 0 24 79) u promenljivu **cEkran**. Ako editanpate ovaj string videćete da se on sastoji od niza parova (znakovi, atribut) Pretkodni poruv veoma oko ekrana u standardnom modu 25x80 simbola.

Ako se kasnije poziva funkcija **RestScreen** na sledeći način

```
RestScreen( 0,0,24,79, cEkran)
```

poseva će se vratiti stari sadržaj ekrana zaštićeni u promenljivoj **cScreen**. Naravno, ovaio operacijom se na uhće na sadržaj promenljive **cScreen**, pa je moguće restaurirati stari sadržaj ekrana proizvoljan broj puta.

Funkcije **SaveScreen** i **RestScreen** se obično koriste u paru, pre poziva ispisa na ekran se pomoću **SaveScreen** sačuva sadržaj ekrana, zatim se vrši željeni ispis i na kraju se pomoću **RestScreen** vraća stari sadržaj ekrana. Tako se posle ispisa poruke određene prirode na ukće na sadržaj ekrana. Evo ispisnog primera kaciđenje.

```
PROC Obrada
   LOCAL cEkran
   // Snimanje ekrana
   cEkran := SaveScreen(0,0,24,79)
   // Ispis poruke
   // koje utiče na sadržaj ekrana
   //
   // Vraćanje starog sadržaja ekrana
   RestScreen(0,0,24,79,cEkran)
RETURN
```

Funkcije sa čuvanje ekrana se obično koriste za realizaci ju raznih sistema menija. Takav sistem čini jedna ili više procedura koje omogućavaju izbor jedne od mogućih opcija Mada deto se zapitati čemu služe takve funkcije pored **AChoice** i sličnih. Pri korišćenju ovih jelementarnih) funkcije za korisnički interfejs brzo dete ustanovati da veoma često ponavljate iste operacije pre i posle poziva funkcije **AChoice**: prevođen gde treba ispisati neku čuvanje sadržaja ekrana i sl Zato je ugodno navedene naredbe objedinit u jedan proceduru i tako znatno pojednostaviti program. Na sledećem listingu je prikazana jedna takva funkcija. Na osnova zadatih koordinata i opcija menija, funkcija **Menu** ispisuje meni i vraća broj izabrane opcije. Kada korisnik završi izbor vraća se stari sadržaj ekrana.

```
PROC Main
Menu 10 0, {"Edit Record", "Delete Rec ", "Add Record "}
RETURN
// Funkcija se izbor
FUNC Menu (nX, nY, aOptione)
   LOCAL cEkran, Xf  XE, Y1, Y2, i, nSelect
   // Koordinate prozora
   X1 := nX
   Y1  := nY
   XE  := nX + Len(aOptione) + 1
   Y2  := nY + Len(aOptione[1]) + 1
   cEkran  = SaveScreen (X1, Y1  XE, Y2)
   @ X1, Y1 TO X2,Y2
   FOR i  = 1 TO Len (aOptione)
      @ X1+i Y1+1 PROMPT aOptione(i)
   NEXT
   MENU TO nSelect
   RestScreen (X1, Xi, X2, Y8, cEkran)
RETURN nSelect
```

Kada treba očuvati sadržaj celog ekrana, koriste se posv **SaveScreen(0 0,24 79)** Međutim treba imati u vidu da *Clipper* podržava razne modove video kartica, pa ekranu može biti različita „dimenzija". Zato je osnovo bolje umesto brojeva 24 i 79 koji predstavljaju koordinate donjeg desnog ugla ekrana u standardnom (25x80) tekst modu koristiti funkcije **MaxRow**() **MaxCol**() koje vraćaju trenutne važeće parametre koordinate donjeg desnog ugla ekrana. Znači, umesto prethodnog. se ću vamo celog ekrana proporučujve je pisati

```
cEkran := SaveScreen (0,0,MaxRow(),MaxCol())
```

Slično se pše i odgovarajuća naredba za restauraciju sterog sadržaja ekrana.

Kada je potrebno sačuvati ceo ekran, posvo funkcije može se upotrebiti i naredba **SAVE** na sledeći način

```
SAVE SCREEN TO cEkran
```

Ovu naredbu protpresseer prevodi u poziv funkcije **SaveScreen**, mke da je svojeobno koji zapa karaktere Kada treba vratiti sadržaj ekrana, koristi se sledeća naredba.

```
RESTORE SCREEN FROM cEkran
```

Može vam se učiniti da je nepotrebno uvek računati koordinata prozora na ekranu čiji sadržaj treba sačuvati – uvek da isti posao obaviti i poziv **SaveScreen** kaji će sačuvati ceo ekran. To je zasvim ispravno ali treba imati u vidu da snimanje celog ekrana traži prilično veliko 25x80x2 = 4000 bajtova, što nije tako malo Kada se snima manji deo ekrana utvrđuje znatno manja je snažno manji. Ova naredba dakazi da uzašje kada je potrebno čuvati sadržaj ekrana vide puta, kada u trvalak memorija može biti veoma značajan

Već smo videli da se funkcije **SaveScreen** : **RestScreen** koriste u paru svakom pozivu funkcija **SaveScreen** (obično) odgovara jedan poziv **RestScreen**. Ovim tegu mnogu se poklapati i koordinate prozora koji se snimljeni Naime, funkcija **RestScreen** nema nikakvu informaciju o tome koji je deo ekrane snimljen. ovim koordinata koje se zadaju kao argumenti Ako se zadaju pogrešne vrednosti slika na ekranu može biti dokru ugnuana. Ovaj efekat se ponekad može i koriano upotrebiti Probajte sledeću proceduru kao semenu se naredbu **RESTORE SCREEN**

```
PROC Rest (cScr)
   LOCAL i
   CLS
   FOR i  = 24 TO 0 STEP  1
      RestScreen (1 i,24 i 79 i,sScr)
      Inkey(0 05)
   NEXT
RETURN
```

Ako se funkciji **RestScreen** zadaju laše koordinate, stara slika neće biti korektno vraćena. Ali ako se zadaju iste ali nemaju prozora, samo na drugoj koordinati, slike će biti dobro vraćena, ali na drugu pomoju na ekranu. Ova osobina je veoma korisna i koristi se za pomeranje objekata po ekranu.

```
#include "Inkey.ch"
PROC Main
   MaveIt (0,0,6,15)
RETURN
// Pomeranje zadatog objekta
PROC MaveIt (X1, Y1  X2  Y2)
   LOCAL cObject, cH  cScr
   cObject  = SaveScreen (X1, Y1, X2, Y2)
   WHILE  T
      cScr  = SaveScreen (X1  Y1, X2, Y2)
      RestScreen (X1  Y1  X2  Y2  cObject)
      Ch  = Inkey (0)
      RestScreen (X1  Y1  X2, Y2, cScr)
      DO CASE
         // Exit
         CASE Ch == K_ESC , EXIT
         // Gore
         CASE Ch == K_UP  AND  X1 > 0
            X1  X2
         // Dole
         CASE Ch == K_DOWN  AND  X2 < MaxRow()
            X1++  X2++
         // Levo
         CASE Ch == K_LEFT  AND  Y1 > 0
            Y1  ; Y2
         // Desno
         CASE Ch == K_RIGHT .AND.
            Y1++ , Y2++
      END CASE
   END DO
RETURN
```

UČITAVANJE PODATAKA

Clipper ima veoma razvijen sistem za učitavanje podataka raznih vrednosti. Razlog je to što je u standardnom *Clipper* programima učitavanje raznih podataka veoma česte operacija.

Najjednostavnije naredbe se učitavanje podataka su **ACCEPT** i **INPUT** Ona su slične naredbama se učitavanje podataka u ostalim programskim jezicima. Pomoću naredbe **ACCEPT** se učitava niz simbola i dodeljuje se zadatoj promenljivi Na primer

```
ACCEPT "Unesi X:" TO nX
```

Kada se izvrši ova naredba, korisnik može da otkuca proizvoljan niz simbola (najviše 255 simbola) koji se posle potvrde (pritisak na taster **<Enter>**) dodeljuje promenljivoj **cX**. Korisnik može da ukuca i broj, ali se tada on mora kadavati. Tada se **SAY..GET** naredbe pše u skraćenom obliku.

```
@ 1,0 GET cScreen
```

Pretkodne **SAY..GET** naredbe još uvek na označavaju unos vrednosti, već samo ograničuje položaja vrednosti. Tak kada se uvrši naredba **READ**, korisnik može menjati vrednosti zadatih promenljivih Kursor se pomera na početak prve vrednosti A ova je označena inverznom bojom (u kontrolu te boje kako reći kasnije) i korisnik može da pomera kursor od jedne do druge vrednosti i menja ih nekom od sledećih komandi

<Up>	Prethodna vrednost
<Down>	Sledeće vrednost
<Enter>	Sledeća vrednost
<Left>	Pomeranje kursora leve u okviru tekuće vrednosti
<Right>	Pomeranje kursora desno u okviru tekuće vrednosti
<Home>	Pomeranje kursora na prvi simbol tekuće vrednosti
<End>	Pomeranje kursora na poslednji simbol tekuće vred.
<Ctrl-Y>	Brisanje do kraja tekuće vrednosti
<Ctrl-U>	Vraćanje originalne vrednosti
<Ctrl-T>	Brisanje reči
<BS>	Brisanje simbola leve od kursora
<Del>	Brisanje simbola na kome se nalazi kursor
<Ins>	Naizmenična promena moda (Insert, O-verzipac)
<Ctrl-Left>	Reč leve
<Ctrl-Right>	Reč desno
<Ctrl-Home>	Prva vrednost koja se unosi
<Ctrl-End>	Poslednja vrednost ili kraj unosa
<PgUp>	Kraj unosa, vrednosti su nemarirane
<PgDn>	Kraj unosa, vrednosti su nemarirane
<Esc>	Kraj unosa, tekuće vrednost se ne memorije

Dajstvo nekih od nabrojanih tastera zavisi i od nekih drugih uslova. Na primer unos se normalno završava pritiskom na **<PgDn>** **<PgUp>** ili **<Enter>** (samo kada je kursor na poslednjoj vrednosti) Međutim, i taster **<Ctrl End>** završa va unos podataka. Ako se želi da unesene naziva promenli funkcija **ReadModal** u datoteci **GETSYS.PRG**

I taster se strelicama **<Up>** i **<Down>** mogu završava ti unos podataka kada se kursor nalazi na prvoj odnosno poslednjoj vrednosti. Ova osobina se ponekad može koriane upotrebiti Da li će ovi tasteri završavati unos kontrolite se funkcijom **ReadExit** koja ima samo jedan argument logičku vrednost koja označava da li tasteri **<Up>** i **<Down>** završavaju unos podataka. Funkcija vraća pretkodne stanje ako je vrašena vrednost T tasteri **<Up>** i **<Down>** su prethodno završavali unos e može nazi Ako, na primer želite da u nekoj vašoj proceduri vaši ova osobina, a da pe završetku procedure ostane prethodno stanje ta procedura se može ovako napisati

```
PROC TestReadi
   LOCAL lVi:=ReadExit()
   ReadExit(.T.)
   ReadModal(GetList)
   ReadExit(lVi)
RETURN
```

Često se naredbe **GET READ** i ostala funkcija koja na neki način služe za unos podataka pa pretkodnom primeru nazivaju **GET** sistem.

KONTROLA I FORMATIRANJE ULAZA

Naredba **GET** omogućava vrlo preciznu kontrolu ulaznog stringa. Ta kontrola se obavlja na nekoliko načina i u različitim trenucima. Kontrola ulaza se obavlja pre unosa podataka, se vremu unosa i posle unosa.

Za vreme unosa se kontrola obavlja na dva načina. Prvi način je vam korisniku korisnika i automatski se određuje na osnovu tipa promenljive koja se očitava. Na primer, pri unosu broja nemate opcije da otkucate slovo, što je razumljivo. Takođe, kad unosite datum automatski će biti postavljeni separatori za određivanje dana, meseca i godine.

Naravno, ovakva kontrola je veoma gruba. Mnogo „finija" načina je upotreba stringove za formatiranje ulaza kao kod naredbe **SAY**. String se formatiranja određuje kako će se tretirati ona će korisnik otkuca. Na primer, ako se nalazi sledeća naredba.

```
@ 0,0 SAY "Iznos: X" GET nX PICTURE "999 9"
```

biće dozvoljen unos samo brojeva sa četiri cifre od čega je jed na decimala. Kada je dejstvo specijalnih simbola našta iznenađenje:

- A Slovo
- N Slovo ili cifra
- X Proizvoljan simbol
- S Cifra
- # Cifra, plus, minus ili blanko
- L T F Y N
- Y Y N
- ! Velike slove
- $ Cifra, $ unesete veličih nula u broju
- Cifra, " unesete veličih nula u broju
  Decimalna tačka
- , Zarez

Sličnu kao kod **SAY** naredbe, postoje i specijalne funkcije

- A Mogu se koristi samo alfabetski simboli
- B Prikazuje brojeve levo poravnato
- C Prikazuje CR posle pozitivnog broja
- D Prikazuje datum u tekućem formatu
- E Menja ulogu zareza i tačke kao separatora
- K Ako se prvo na prilazne neki od tastera za kursorisanje, briše se inicijalna vrednost
- R Slabiji koji nisu specijalni se prikazuju pri unosu, ali se ne dodaljuju gramatičkoj
- S Skrolovanje vrednosti koja se unosi na N pozicija
- X Prikazuje se DB posle negativnog broja
- Z Prikazuje vuln kad blanke
- (Zatvara negativan broj u zagrade
-) Zatvara negativan broj u zagrade bez vodećih blanka simbola
- ! Sva slova se konvertuju u velika

Na primer na sledeći način se može unositi telefonski broj kao string

```
@ 0,0 SAY "Telefon:" GET cTel PICTURE "@K (999)999-
999"
```

Prethodna naredba specifiše unos poziva broja i šestocifrenog telefonskog broja. Funkcija **K** označava da se briše prethodna vrednost ako je prvi pritisnut taster se cifrom.

Kontrola ulaza se još može obaviti pre i posle unosa podataka. Kontrola pre unosa se obavlja opcijom **WHEN** i pomoću nje se kontroliše da li će se uopšte dati promenljivi unositi obaviti. Takve situacije se akvilja kod unosa grupe podataka, kada je neki podatak zavistan od ostalih. Na primer ako se u neke lični podaci za jednu osobu, na unosi spomen pola mogu se selektivne unositi ostali podaci (kod ženske osobe nema smisla pitati za vojni rok)

Opcija **WHEN** prihvata logički uslov ako je uslov zadovoljen, unosi se vrednost takođe promenljive, u u suprotnom se preskoči na unos sledeće promenljive. Na primer

```
@ 0,0 SAY "Ime i prezime" GET cPr
@ 1,0 SAY "Da li je u braku" GET lUBraku PICTURE "
@ 2,0 SAY "Ime bračnog druga" GET cPar WHEN lUBraku
@ 3,0 SAY "Adresa" GET cAdresa
READ
```

Ako se se logičnu promenljivu **lUBraku** unese vrednost .T (osoba je u braku) kursor se pomera na vrednost promenljive **cPar** gde treba unati ime bračnog partnera. Ako je vrednost promenljive lUBraku = .F (osoba nije u braku) tenno prelazi odmah na unos bračnog partnera, pa se kursor pomera na vrednost **cAdresa**.

Kao **WHEN** uslov se može zadati proizvoljan logički uslov, sadi i poziv neke funkcije koje vrati logičku vrednost. To se najčešće koristi za automatsko otvaranje menija se mogućim opcijama.

Slično **WHEN** uslovu postoji i **VALID** uslov koji omogu ćava da se kontrola izvrši posle unose vrednosti. Na primer ako je potrebno uneti broj vedi od 0..to se može obezbediti na sledeći način

```
@ 0,0 GET nX VALID nX > 0
```

Ako korisnik unese korektnu vrednost, unos se normalno nastavlja, jer je uslov ispunjen Ako se unese vrednost takva da je uslov netačan, korisnik se može pomeriti kursor ove dok se unese korektana vrednost ili odustane od unosa tasterom **<Esc>**

NAREDBE ZA KONTROLU TASTATURE

Kada korisnik pritisne neki taster kod pritisnutog tastera se upisuje u posebnan bafer. Kodovi se upisuju onim redom kako korisnik aktivira određene tastera. Na primer ako korisnik pritisne tastere **A, B** i **C** u bafer tastature biće upisani sledeći kodovi.

```
<A> <B> <C>
```

Kodovi tastera se u *Clipper*-u nazta realizuju od standardnih kodova. kodovi svih simbola iz *ASCII* tabela su isti, dok nagotivni i kodovi veći od 255 odgovarajiu funkcijskim tasterima, **Ctrl** i **Alt** kombinacijama.

Svaka od naredbi koja koristi tastaturu se interaktivnu se korisnikom šalju svoje upisane podatke u bafera i iz onim redom neke su upisani snadi prve te biti pročitan kod simbola **A**, tid **Kada se kod tastera pročita**, na prvo mesto za obradu dolazi kod sledećeg pritisnutog tastera. Funkcija i naredbe **READ**, **AChoice**, **DBEdit** i sl. automatski obavljaju ovu postupak i programer osnu potrebe da utiče na njega osim u nekim posebnim prilikama)

Sinanrage tastature se obavlja preko interrupta – to znači da ako na vreme nake duže obrade pritisnete jedno ili više tastera, oni se biti unesni u bafer tastature, all se biti obrađeni tek kad se tekuća obrada završi i neki *Clipper*-ov sistem naredbi sledeći kod tastera.

Osnovna funkcija preko koje programer unima kodove iz bafera tastature je funkcija **Inkey**. Ova funkcija vrati kod tastera iz bafera tastature koji treba sledeći obradit. Ova veoma važna funkcija se može podivati na tri različita načina.

Prvi je način da se funkcije pozove bez argumenata i tada vrati kod pritisnutog tastera ako nešto postoji u baferu tastature ili 0 ako je bafer tastature prazan (korisnik nije pritisnuo ni jedno taster) Ovo se najčešće koristi prilikom dugih obradi podataka, kada korisnik pritiskom na neki taster označava da želi da prekine tekuću obradu.

```
WHILE .T.
  ? "Unesite ime"
  IF Inkey() = 27
    EXIT
  END IF
  SKIP
END
```

U ovom primeru se ispisuju redom slogovi sve dok se ne dođe do kraja datoteke ili dok korisnik ne pritisne neki taster Ako korisnik ne pritisne nijedan taster za vreme obrade, funkcija **Inkey** ta uvak vraćati vrednost 0 pa će se cikina normalno nastaviti Vidine je da uvđite da u svom slučaju funkcija **Inkey** na čeka da korisnik pritisne taster – ona jednostavno vrati tekuće stanje.

Mnogo češće se koristi varijanta u kojoj funkcija **Inkey** čeka da korisnik pritisne taster Teda se funkcija **Inkey** poziva se argumentom koji označava koliko sekundi treba čekati na akciju korisnika.

```
? "Pritisni bilo koji taster"
? "Prekidanje za 10 sekundi"
```

U ovom primeru funkcija **Inkey** čeka 10 sekundi da korisnik pritisne neki taster Ako za 10 sekundi korisnik na pritisne ni jedan taster funkcija vrada vrednost 0 U suprotnom vrada kod pritisnutog tastera. Naravno ako u trenutku poziva bafer tastature već nadrži kod nekog tastera, funkcija odmah vrada vrednost pritisnutog tastera. Znači, može se reći da **Inkey** fa na pritisnuti najviše 10 sekundi Ovo se obično koristi za upisivanje ograničenog broja znakova unesna na ekranu a zatim se briše. Ako korisnik pročita poruku pre predviđenog vremena, može nastaviti rad jednostavnim pritiskom na neki taster

```
? "Pritisni odgovarajuće slovo"
? "Prekidanje za 10 sekundi"
```

Treći način korišćenja funkcije **Inkey** je u situaciji da se čeka sve dok korisnik ne pritisne neki taster tada se kao argument zadaje vrednost 0. Na primer

```
? "Pritisni odgovarajuće slovo za nastavak"
Inkey(0)
```

Funkcija **Inkey** vrada kod tastera koji je na vrdu za obradu kada ova funkcije vrati kod tastera. on se briše iz bafera tastature Vovom slično reda i svi ostali sutemi koji koriste tastaturu. Na primer naredba **READ** koristi funkciju **Inkey** za skaniranje tastature Poše su neki od svih sistema setverzal su korisnike (**AChoice**, na primer), nekad je korisno znati koji je taster **poslednji** unet iz bafera tastature. Za to se koristi funkcija **LastKey** koja vrada kod poslednje obrađenog ta stera. Na primer kada se vrši unos podataka naredbom **READ** u memorijske promenljive, i kada se unos završi pritiskom na taster **<Enter>** treba upisati podatke iz memorije u datoteku, a ako korisnik pritisne taster **<Esc>** na treba upisati podatke Nevolja je u tome što je taster koji je završio unos već obrađen, pa je izladen iz bafera tastature Teda se koristi funkcija **LastKey**

```
@ 0,0 SAY cAdresa
@ 1,0 GET cAdresa
@ 2,0 GET Gota_O
READ
IF LastKey() == K_ESC
  RETURN(.F.)
END IF
```

Poslednji je zgodno videti koji je taster na redu za obradu ali ga ne uzimati iz bafera tastature Za to se koristi funkcija **NextKey** Ona radi isto kao **InKey** bez argumenata se razli kom što ne pri uzimanju sledeđeg tastera iz bafera tastature no ukianja kod iz bafera. Korišćenjem funkcije **NextKey** možete

možno pojednostaviti neke situacije svojevrsnim „survivingom u budućnost"

Veličina bafera tastature se može podešavati naredbom **SET TYPEAHEAD TO** n, gde je N neva veličina bafera tastature Ukoliko se veličina postavi na 0, svaki pritisnut taster se upisuje samo u bafer koji koristi *BIOS* za akumuliranje tastature.

Naredba **CLEAR TYPEAHEAD** briše bafer tastature Ova naredba se koristi kod je potrebno sprečiti da neki znakovi ka kod tastera u baferu rezultuje neželjenom akcijom. Na primer

```
? "Da li ste sigurni da brišete trenutnu datoteku"
// Očekuje bafer tastature
CLEAR TYPEAHEAD
IF Char(Inkey()) == "D"
  DELETE FILE
END IF
```

Kod programiranja korisničkog interfejsa strukturna programa prati akcija korisnika. kada korisnik treba da pročitan neku akciju onda se akanura tastature, kada korisnik posavlja akciji pojavljuje se ciklus, tid Ponašanj akcije korisnika mogu veoma zakomplikovati program jednostavno se nura predvideti obrada svih mogućnosta koje korisnik ima. Kada ki korisnik mrae baviva je strukturna programa, onda bi posao programera bio mnogo lakši Pošto obično nije tako, programer mora ponekad i narušiti strukturu programa da bi „prešao" korisniku. Pri tome jedna naredba ima značajnu ulogu **KEYBOARD** Ona briše bafer tastature i upisuje zadati niz znakova u njega, kad kad će je korisnik otkucao na tastaturi taj niz znakova. Naravno, funkcija **InKey**, **LastKey**, **NextKey** kad i ostale se prave nikakvu razliku između onog šta je korisnik otkucao i rezultata **KEYBOARD** naredbe. Teko se za idealan način korisnik privremeno zamenjuje „idealnim" korisničkim koji „zna" šta treba da pročita i dalem tranzakcu.

Naredba **KEYBOARD** ja veoma jednostavna, ali iskorišćenom i veoma moćna. ona vam omogućuje da iznenada tačno se programiranje podvedete pod opšta pravila. Nije laka na jednostavnom primeru pokazati moć ova naredba ali je sigurna da će je veoma često koristiti.

KEYBOARD naredba prihvata samo hodove tastera i vrednu 0 i 255 – te znači da nje moguće direktno upisati kod funkcijskog tastera u bafer tastature Na primer, sledeća **KEYBOARD** naredba unosi vrednost prve **GET** promenljive i polaje nam od drugog polja.

```
KEYBOARD "Petar Petrić" + Chr(13) + Chr(13)
@ 0,0 GET cPrezime
@ 1,0 GET Adresa
@ 2,0 GET Broj
READ
```

Clipper nabrzi još jednu naredbu sličnu i moćnu naredbu za rad se tastaturom. **SET KEY** Ova naredba omogućava da se definiše procedura koja će se aktivirati uvek kada se pritisne neki taster na primer pritiskom na taster **<F1>** dobijaju se kaje informacije Taster **<F1>** se automatski definiše kad help taster na potrebu programa. U stvari, *Clipper* automatski izvršava naredbu

```
SET KEY 28 TO Help
```

sime se tastera **<F1>** (kod ovog tastera je 28) dodaljuja procedura **Help** Od izvršenja ove naredbe, pritisak na taster **<F1>** pozva proceduru **Help**. Pri tome se automatski proslediju tri parametra naziv procedure u kojoj je aktiviran taster broj kaje tekste programa (ako postoji ova informacija) i sadržaj promenljiva koja se trenutno uređava.

Naredba **SET KEY** ima ograničenja. njena definicija vezi samo dah program šake na jedan na primer u **READ**, **ACCEPT** **AChoice** **MemoEdit** itd. Jedini izuzetak je **InKey** – pri izvršenju ove naredbe se na uzimaju u obzir tasteri definisani se **SET KEY**

Sasvim slično tasteru **<F1>** koga *Clipper* automatski definiše, može definisati i vaše tastere. Na primer

```
Kalkulator Trans.prg
//
SET KEY K_ALT_K TO Kalkulator
```

Pasle sve naredbe svaki prijanak na taster **<Alt-K>** izazvaće i izvršenje procedure **Kalkulator**. Sva tri opisana parametra – nazivi procedure koju lokiju i nazivi promenljivih se automatski proslеđuju. Ove parametre (kao i teke druge) ne treba navoditi u naredbi **SET KEY**. Ukoliko procedura ne postoji naredbe **SET KEY** a neće nikakvog efekta.

BOJE

Važan detalj pri izradi programa predstavljaju boje *Clipper* omogućava vrlo dobru kontrolu boja i to svakako treba što je moguće više iskoristiti ukoliko i nenamenjive „oterasan" program ima znatno predniji nad crno-belom varijantom. Naravno pod „crno-bele" se ne podrazumeva varijanta za monokromatski monitor – i ona mora biti osvešćena bojama, u grani samo mogućnosti *Clipper* poznaje sedmiku osnovnih boja, u sevmenti od tipa grafička kartice:

BOJA	OZNAKA
CRNA	N
BELA	W
PLAVA	B
ZELENA	G
CRVENA	R
CIJAN	BG
LJUBIČASTA	RB
BRAON	GR

Naravno, za monokromatskom monitoru se mogu koristiti samo boje i crna boja. Kao dodatak, može se uključiti atribut podvlačenja znakom **U**. Za svaku od ovih boja postoji i svetlija varijanta, boja se označava plusom **B+** označava svetlo plavu boju **GR+** svetlo braon (žuta) tkl. Može se koristiti i oznaka ***** koja uključuje treptanje. **G*** označava zelenu boju koja treperi

Da bi se boja promenila, mora se zadati nove boje slova i boje pozadine. Ta se radi konstruisanjem **2b** gde je **F** boja slova, a **B** boja pozadine. Kada se promeni boja, ova ispisivanja na ekranu će se nadalje obavljati izabranom bojom. Za promenom aktivnu boju koristi se funkcija **SetColor** koja postavlja novu boju slova i pozadine. Na primer **SetColor ("w/r")** uneću boju u kojoj slova su crvenog pozadinu. Stvar ipak nije tako jednostavna. Pošto ovi podaci*Clipper*a podnju podatke u tekstualnoj boji na bi bilo spadno da se ove ispisuje istom kombinacijom boja. Zato se pri zadavanju nove boje mora zadati 5 kombinacija raadvojenih zarazima. Na primer

```
SetColor ("w/r,n/w,,,n/w")
```

Prva boja je zaysiinaja i koristi se za sve funkcije koje ispisuju vrednosti na ekran. **SAY QOut, @..TO** Za pretkodni primer je to **W/N** – bela slova na crnoj pozadini.

Druga boja se koristi za izabrane elemente menija i izabrane vrednosti u **GET** sistemu – ova boja obavezno postavite da bude različita od prve, kako bi korisnik jasno video šta unosi.

Treća boja označava boju okvira – ova boja nas zasluja samo na VGA karticama.

Četvrta boja se ne koristi i može je proskočiti pri specifikaciji

Peta boja se odnosi na neodabrane elemente menija i vrednosti u **GET** sistemu. One se obično postavlja da budu i sta ili slična kao prve boje.

Poslednji oblikas prave ogromnu greškU i funkcija **SetColor** koristе pa potrebi u nekom programu. Ta resuškuje stvaranjem tekućeg pu pokušaju da se promene boje u programu. „Jurnji" definicije boja pa programu nije nimalo zabavan posao Osnovno pravilo je da se sve boje koje se koriste definišu na početku programa, zatim bi se kasnije izmene vršile na jednom mestu Dobre je predvideti da će program raditi i na različitim tipovima monitora Tip monitora (koler ili monokromatski) može se ispitati funkcijom **IsColor()** koja vraća .T. ako je na računaru koler grafička kartica. Ovdno se, u zavisnosti od tipa monitora, koriste jedan od tipova **IsColor** postavljaju odgovarajuća vrednosti boja. Na primer

```
IF IsColor()
   Normal := "w/b, n/w, , , n/w"
   MenuIC := "r/bg"
   BtnIC  := "w/n, n/w, , , n/w"
ELSE
   Normal := "w/n, n/w, , , n/w"
   MenuIC := "n/w, , , , n/w"
   BtnIC  := "n/w, , , , n/w"
ENDIF
```

Obično se koristi prva boja u definiciji, tako da ponekad nije važno postaviti sve boje.

Gotove sve naredbe ne sepu na ekran posebnu mogućnost privremene promene boje koja se menja samo za tekući ispis, a dalje ostaje da vaši tekuće definicija. Na primer naredbe @ 0,0 **SAY "Clipper" COLOR "w/b"** ispisuće poruku belim slovima na plavoj pozadini bez obzira na tekuću boju.

RAD SA BAZOM PODATAKA

Baza podataka je, pojednostavljeno gledano, veliki skup organizovanih podataka koji se odnlava za pomoć računara. Ta mogu biti razna podaci' teka mogu postojati baze podataka o zaposlenim radnicima jedne firme: o knjigama i članovima neke biblioteke itd Svi te podaci se na neki način organizuju a posebna program (koji se često naziva **sistem za upravljanje bazom podataka**) omogućuje korisničkima pristup do podatka. Taj program omogućuje npr novih podataka, pregled i izmenu postojećih, a za vre to brzo a sigurnosti tih podataka, kako bi korisnike bez i pod kojim uslovima pristupa bas, sta ra se da se ranaim izmenama ne dođnju pogrešni podaci itd. Može se reći da je *Clipper* specijalizovan programska jezik namenjen pisanju programa koji upravljaju bazom podataka. Ta „specijalizovanost" se ogleda u tome što *Clipper* ima posebne naredbe koje omogućuju jednostavnji rad se podacima u bazi Baza podataka bojima upravlja program napisan na *Clipper*u je, u stvari skup dste teke. Svaka datoteka sadrži neke podatke, o sve zajedno čeka baru podataka.

Clipper koristi takozvani **relacioni model** baze podataka. U relacionom modelu se realni objekti i njihovi odnosi predstavljaju preko relacija odnosno tabela. Svaka od tih tabela predstavlja jedan objekat. Na primer tabela učenika može imati sledeću strukturu

IME	DOBIJE	ŠIFRA
Pera Perić	345/90	1002
Mika Milić	120/91	1003
Zora Zerić	342/90	1004

Kolone su **IME i PREZIME DATUM ROĐENJA, RAZRED** Svaki red sadrži podatke se jednog učenika. Kada treba spisati podatke se novog učenika, upunje se novi red i u njemu odgovarajuć podaci, a zka treba dodati novi podatak na sve učenike (**USPEH**, na primer) dodaje se nova kolona.

Ovakav način organizovanja podataka je vrlo jednostavan, pregledan i teko shvatljiv **Naziv relacione base podataka** dolazi od matematičke osnove na kojoj je zasnovana ovakva organizacija. Naime, u matematici se koristi pojam **relacija** se određeni skup uređenih n-torki i što se može najjednostavnije


```
Kalkulator Trans.prg
//
SET KEY K_ALT_K TO Kalkulator
```

je predstaviti tabelom jasa na slici Kada se da je tom tabelom definisana relacija **UCENIK** koja je tačna se svaki red Na primer n-torka

```
("Pera Perić", "5/6", 345/90, 1002)
```

pripada relaciji, jer je Pera Perić učenik reden 12 04.82 i ide u 7 razred.

Tabela vrsna podsećaju na klasične datoteke svaka kolona može biti jedno polje a svaki red jedan slog u datoteci. U stvari, u *Clipper*-u je to upravo teko organizovane svaka tabela/relacija se predstavlju jednom datotekom u kojoj su kolone polja, a redovi slogovi Ova je sistem se predstavljanja tabela koji se koristi u svim profesionalnim bjh sistemima, kompatibilnim se *Dbase-om* (i *Clipper* spada u te pro svoda) Međutim, to ju samo jedan od načina. mnogo relacione baze podataka ne koriste ovakav sistem i mnguća su razna rešenje Na primer ava tabela se mogu čuvati i u jednoj datoteci Zbog prethodno opisanog razloga, termini **tabela, red, kolona** se vrsnu često u *Clipper*-u zamenjuju terminima **datoteka, slog, polje**. U originalnoj literaturi koju se dobiju uz *Clipper* koriste se termini **database file, record i field**.

Već je rečeno da *Clipper* sadrži specijalni skup naredbi i funkcija koju omogućavaju rad sa tabelama odnosno datoteka ma. Na početku rada se takvim letjelom potrebuje izvršiti **otvaranje**, a na kraju **zatvaranje** Datoteka se otvara naredbom **USE <naziv datoteke>** o satvaru naredbom **CLOSE <naziv datoteke>** Zatvaranjem je od slogova jedinaka dužina i struktura Tekuće datoteke imaju ekstenziju **DBF** (od **Data Base File**) kada se koristi relacioni *Clipper* sistem. Na primer otvaranje **DBF** datoteke **PRIMER.DBF** otvaruje se na redbom

```
USE PRIMER
```

Ekstenzija nije potrebne navoditi Ukoliko naziv datoteke koje se otvara nije poznat se vreme pisanja programa, koriste se takozvani „podrazumevani izraz" te je u stvari izraz koji vraća string. Pale se zatvoren u male zagrade

```
USE (Ovaimo)
```

Ako se napiše **USE "Primer"** to će korektno raditi ali ako se napiše neko komplikovaniji izrna (na primer **USE "Pri" + "mer"**) prevodilac će javiti grešku

OSNOVNE OPERACIJE SA DBF DATOTEKAMA

Svaka **DBF** datoteka se pru koriđenju mora **kreirati**. Pri kreiranju se zadaje kakva će biti struktura sloga. Ij koja će polja ulaziti u slog Standardna *Clipper*-ove datoteke se sastoje se slogova jednake dužine, pri čemu je struktura svakog sloga i sta i unaprsd definisana. U terminima tabela, pri kreiranju se zadaju kolone koje će ući u tabelu

Za svako polje u slogu se mora odrediti tip podataka koji će se u njega upisivati Ovde ne vaši fleksibilnost *Clipper*-ovih dinamičkih promenljivih u polje numeričkog tipa se mogu upisivati tekstualne brojevi! Polja mogu biti tipa string, broj datum logičke i memo polje. Stringovi brojevi datumi i logičke vrednosti su u poljima tvrtnuju na sasvim određene načina. Jedino ograničenje je da se se vre tipove mora unaprsd zadati maksimalna veličina podataka koji će biti upisivani Tako na primer se polje tipa string se zadaje maksimalan broj simbola u stringu, se brojeve ukupan broj mesta na prikazivanje broja i broj decimala. Mnoga polja se naše specifičnije i služe se memorisanje vešlh stringova (do 64K) Ovakva polje se ne pamti u osnovnoj datoteci, već se po potrebi upisuje u posebnu datoteku (se standarni drajver se ovi podaci upisuju u datoteke se ekstenzijom **DBT**) To znači da je moguće u svakom slogu bezlati string promenljive dužine pri čema nije potrebno una pred fikano alocirati prostor osga se sa koristi po potrebi lnade, ovakva polja se obrađuju kao i klasični stringovi

Datoteka se kreura u dva fase: prvo se u jedan niz upiše opis strukture datoteke. Tu se upisuju nazivi polja, tip i dužina xaje, a zatim se pozova funkcija se kreiranje datoteke:

```
DBCreate ("a", {;
   {"Ime",  "C", 20, 0},;
   {"Adr",  "C", 40, 0},;
   {"Mat",  "N", 10, 2},;
   {"Ocena", "C", 1, 0};
})
```

Pri zadavanju polja treba pasle tipa polja zadati i dva vrednosti a ikrnu polja (koja zamaju svak osnovu za primer kod stringova je važna samo prvi broj dok je drugi bez značaja) ali ih je obavezna navesti ux ove lopove: ukoliko se ne koristi treba navesti 0 Kao zanacrtnih polja se zadaju dva vrednosti. ukupna širina polja i broj decimala. Funkcija **DBCreate** ima dva argumenta. Prvi je naziv datoteke koju treba kreirati (ne treba navoditi ekstenziju), a drugi je niz koji sadrži strukturu sloga.

Kada se kreira **DBF** datoteka, *Clipper* podržava razne operacije na njenom slogovima. Kao što je već rečeno pre bilo koja operacije, datoteka treba otvoriti. Za to se koristi naredba **USE**. Na kraju rada se datoteka zatvara naredbom **CLOSE**

```
USE Test
Rad sa datotekom
CLOSE
```

Šali kao i kod standardnih datoteka, in datoteke se mogu čitati tli u nju upisivati slogavi Ova operacija se obavljaju automatski i programer ne mora da vodi računa o tome. Obrada se u vek vrši nad **tekućim slogom**. Koji će to slog biti određuje se specijalnim pokazivačem sloga, koji se vodi za svaku otvorenu **DBF** datoteku. Ovaj pokazivač sadrili broj sloga koji se trenutno obrađuje. Kada se datoteka otvori leka 5 slog je slog broj jedan.

Ako želimo da radimo se nekim drugim slogom, koristimo naredba **GO** koja pomera pokazivač sloga na slog čiji broj zadamo Na primer naredba **GO 220** pomera pokazivač sloga na slog broj 220 Pasle izvršene ove naredba možemo izvršiti ra ma operacije se ovim slogom. Naredba **GO** ima i dva dodatna oblika. prvi je **GO TOP** koji pomera pokazivač na prvi slog u datoteci a drugi je **GO BOTTOM** koji pomera pokazivač na poslednji slog u datoteci.

Često je potrebno proći na sledeći slog. Na primer, ako je pokazivač sloga na slogu broj 220, treba izvršiti naredba **GO 221** Ta je mnogo bolje uraditi naredbom **SKIP** ona služi za pomeranje pokazivača sloga na prvi naredni slog datoteka. To se obično koristi kad je potrebna uraditi neka operacija redoma se svim slogovima u datoteci.

Naredba **SKIP** se može koristiti za pomeranje na prethodni ili slog proizvoljna „skoljen" od tekućeg Na primer **SKIP 10** „preskače" naredni 10 slogova i pokazivač pomera za deset slog Ukoliko se pomeranje vrši unazad koriste se negativni brojevi. **SKIP -1** pomera pokazivač na slog pre tekućeg.

Kada se pretkodno naredbama kakvo određeni slog u datoteci mogu se vršiti razne operacije se njim. Osnovna operacija se očitavanje sloga u memoriju i upisivanje podataka iz memorije u slog Ova posso se u *Clipper*-u obavlja automatski kad god pomerite pokazivač sloga, očitava se tekući slog u operativnu memoriju Tako će se pasle naredbe **GO 220** u memoriji naći podaci iz sloga broj 220 Ti podaci su programu dostupni na krajnje jednostavan način podaci se očitavaju a specijalne memorijske promenljive kojima se pristupa preko imena polja u **DBF** datoteci No primer ako je jedno polje u datoteci **Datum**, vaš program može raditi na fiktivnom memorijskom promenljivom koja se također zove **Datum**. Ako se i zvrši naredba

```
? Datum
```

odštampaće se vrednost polja datum u tekućem slogu Ako se izvrši naredba

polje **Datum** u tekućem slogu dobiće vrednost koja odgovara sistemskom datumu. datumu iz Vesma slična pravila vaše i se ostale ključne polja. Na taj način možete utkask da direktno radite se podsistema u datoteci, što vasema olakšava programiranje.

Treba imati stalno na umu da vrednost svakvih promenljivih direktno zavisi od pozicije pokazivača sloga, što se vidi i iz sledećeg primera.

```
USE Test
? Datum                    && Ispisuje NIL
DO 10
? Datum                    && Ispisuje 04.04.90
SKIP
? Datum                    && Ispisuje 05.05.90
```

Postavlja se pitanja šta se dešava ako se pokuša pomeranja pokazivača sloga izvan granica datoteke? Na primer, zada se naredba GO 3000 a datoteka ima samo 2000 slogova. Ako se pokuša pomeranje pre početka datoteke operacija jednostavno neće uspeti i pokazivač se pomeri na prvi slog datoteke. Pokušaj pomeranja na poslednjeg sloga takođe neće uspeti, ali se pokazivač pomeri za tzv. „fantomski" slog koji se nalazi iza poslednjeg sloga, ima istu polja kao i ostali slogovi u datoteci, ali su sva polja prazna i njihova vrednost se ne može menjati. Kada se pokuša se pomeranjem pokazivača sloga pre početka naredbom **SKIP** funkcija **BoF** vraća vrednost **TRUE** i tako analogna početak datoteke. Na primer, sledeći program obrađuje sva slogove od tekućeg do prvog:

```
USE Test
DO WHILE ! BoF()
   SKIP -1
END DO
```

Pomeranje pokazivača ina kraja datoteke detektuje se funkcijom **EoF**. Standardna konstrukcija za obradu svih slogova u datoteci je,

```
DO WHILE !EoF()
   ...
   SKIP
END DO
```

Eva kako izgleda inicijalizacija sadržaja tabele učenika.

```
PROC Main
   USE Ucenici
   WHILE ! Nof()
      ? Prezime, Ime
      SKIP
   END DO
   CLOSE Ucenici
RETURN
```

Ovaj program inicijalizuje redom sve slogove **DBF** datoteke **UCENICI**. Funkcija **Eof** inicijalizuje da b je pokazivač sloga datoteke došao do kraja. U ovom primeru sme pretpostaviti da svaki slog datoteka **UCENICI** sadrži polja **Ime** i **Prezime**. Da bi se pomenuta vrednost polje u slogu može se koristiti naredba za dodeljivanje = (ux odgovarajuća ograničenja) Međutim mnogo je preporučljivije za te koristite naredbu **REPLACE** koja ima potpuno isto delovanje ali je „neličnije" u programu, pa se tako ravnjikuje dodeljivanje vrednosti standardnim memorijskim promenljivama od upisa u datoteka. Na primer naredba **Datum := Date()+1** može se napisati i kao

```
REPLACE Datum WITH Date()+1
```

Ako je potrebno odrediti koji je broj trenutnog sloga, koristi se funkcija **RecNo()** una vrsdu rednu broj tekućeg sloga. Funkcija **LastRec()** vraća broj slogova u otvorenoj datoteci.

ORAJVERI ZA PRIETUP PODACIMA

Sistemo praka bege *Clipper* pristupa tabelama/datotekama je izmenjiv Naime *Clipper* podržava pristup tabelama napisanim u raznim formatima: o **DBF** datoteke ou samo jedan od takvih formata. Na primer *Clipper* program može priknpsati tabelnom zapisanom u formatu *Borland Paradox* bez ooskakka. Za svaki format je zadužen poseban drajver (**RDD** – **R**eplaceable **D**atabase **D**river). Na primer se prnsiup *Paradox* tabelama zadužen je **DBFX** drajver za standardne **DBF** datoteke **DBFNTX** drajver. Svaki drajver radi potpuno „nevidljivo" se programerove i korisnika, tj. program se ne mnje, pnkna je potrebnom registrnti tip datoteke prilikom otvaranja. Za to se koristi opcije **VIA**.

```
USE Ucenici VIA "DBFNTX"
```

Ova naredba otvara *Paradox* tabelu **UCENICI**. Dalji rad je potpuno isti kao da je reč o klasičnoj **DBF** datoteci. ocbna naredbi i funkcija fe se ponašati zahtijeva Međutim neke razlike ipak postoje. Na primer, *Paradox* tabele nemaju rednu brnj sloga (ali tnte mogu sa svaki slog vrsdnost primarnog ključa!) pa naredba **RecNo()** neće vratiti redni brnj tekućeg sloga već vrednost primarnog ključa. Analogno i naredbe **GO** kao argument u evem slučaju ima vrednost ključa.

Us *Clipper* se dobijaju četiri **RDD** drajvera. **DBFNTX** (standardni *Clipper* format), **DBFNDX** (*dBASE III*) **DBFCDX** (*FoxPro*) **DBFMDIX** (*dBASE IV*) i **DBFX** (*Paradox*) Na tržištu raznih biblioteka se *Clipper* postoji još drajvera. Svaki drajver je, u stvari jedna biblioteka koja se po potrebi uključuje u **EXE** program. Ukoliko želite da u progrnt nu koristite neki drugi format podataka od standardnog trebate vas početnoj programa uključiti da se on nkoličko koristiti drugi drajver funkcijom **RddSetDefault**.

```
RddSetDefault( "DBFNTX" )
```

Sada će podrazumevanu drajver biti **DBFX** i vsi program će (skoro) bez ikakvih izmena koristiti *Paradox* kompatibilan sa pa podatnka. U okviru istog programa možete paralelno koristiti više drajvera – potrebne je samo upotrebiti opciju **VIA** pri otvaranju tabela Na primer:

```
USE Ucenici VIA "DBFNTX"
USE Artikli VIA "DBFNTX"
USE Partneri VIA "DBFX"
```

otvara **DBF** datoteku **ARTIKLI** sa karakteristikama *standardnim* **DBFNTX** drajverom, a datoteku **PARTNERI** kao *Paradox* tabelu .Naredbe su obradu u programu su dalje potpuno iste kao da se koristi standardni drajver.

Neki od nabrojanih drajvera imaju dodatne osobine koje za postaje kod drugih Na primer **DBFCDX** podržava višestruke indekse (o tema nešto kasnnje) Zate se neke naredbe mogu koristiti samo ako je aktivan određeni drajver U suprotnom se dobija poruka o grešci.

Pri kreiranju datoteke se u funkciji **DBCreate** može navesti i treći parametar i an ada sezadava drajver koji se koristi pri kreiranju datoteka. Na primer sledeća naredba kreira*Paradox* tabelu.

```
DBCreate( "Ucenici", aStruct, "DBFX" )
```

Ukaška se ovaj parametar sa navode, koristi se trenutno aktivan drajver

RADNE OBLASTI I ALIASI

Dnne je videti da su vrednosti u tekućem slogu dostupne ime dnkvog eksplicitnog ukazivanja u memoriju. Postupak je potpuno automatski i programer za iepha da radi brigu o tom. Pošto se obično radi se više **DBF** datoteka (odnosno tabela) komunikacije između svake datoteke i programa obavlja se pomoću specijalne memorijske oblasti koje se nazive **radna oblast** ili **radna zona**. Svakoj otvorenoj datoteci se pridružuje jedna takva radna oblast u kojoj se pamte karakteristični podaci za tu datoteku. nesiv vrednosti polja tekućeg sloga, pokazivač tekućeg sloga, itd. Vrlo je važno znati da se otvaranju svake datoteke dodeljuje njena radna oblast. U programu se ova naredba odnosi **ne** tekuću (aktivnu) radnu oblast osim kada se eksplicitno na nagiasi operacija u drugu oblasti Kada se datoteka otvara, ona se uvek vezuje se trenutno aktivnu radnu oblast. Kada program počne se izvršavanjem aktivna je **prva** radna oblast .Pre nego što se otvori naredna datoteka potrebno je aktivn.ste novu radnu oblast u suprotnom se prethodno otvorena datoteka zatvara. Ova se često zaboravlja pa može nepravilno prilično veliko problema Radne oblasti su numerisane brojevima od 1 da 240 a određivanje trenutne aktivne obavlja naredba **SELECT <broj oblasti>** Na primer

```
PROC Main
   // Aktivira se prva oblast
   USE Mesta

   // Aktivira se druga oblast
   SELECT 2
   USE Predsavci

   // Ponovo se aktivira prva oblast
   SELECT 1

   // Oslobađaju se obe radne oblasti
   CLOSE Mesta
   CLOSE Predsavci
RETURN
```

Zapamtite da u jednoj radnoj oblasti istovremeno može biti otvorena samo jedna datoteka ako pokušate da otvorite sledeću, prethodna će se zatvoriti

Pažte pa rad so brojevima kao identifikatorima oblasti vasma neugodan prepucljivije je koristiti tzv **alijasa** Alias je simbolično nazov radne oblasti pomoću koga možemo pristupa ti oblasti bez korišćenja broja, već simboličkim imenom. Na primer umesto **SELECT 3** n prethodnom primeru možemo pisati **SELECT Predsavci** Naravno, talov za promjenjive radnog oblasti pomoću alijasa je da je n njoj otvorena datoteka se tne nazivom Nalnse kada se otvori datoteka u neku radnoj zonu, te radna zona automatski dobije alias kaji je isti kao i naziv otvorene datoteke

Pažte osnm datoteka mogu biti prilične negrakičnа, mopde je uvesti naziv radne oblasti korišćenjem opcije **ALIAS** prilikom otvaranje datoteke Na primer

```
SELECT 37
USE Predsavci ALIAS Podaci
```

Sad su naredba **SELECT 87** i **SELECT Podaci** potpuno rav nopcavna ali je najbolje koristiti drugu verziju. Pogrešno bi bilo pokušati se **SELECT PRFOICOS** jer opcija **ALIAS** pomiiuje naziv datoteke kao alias Za odredivanje aliasa radne oblasti se koristi funkcija **Alias**. ona vraća string koji sadrži uvabolički nazov tekuće radne oblasti Na primer

```
? Alias(37)    // "PODACI"
```

Početnici vesma često zapreve skrku se slucsma i radnim zonama n programu, tako da navodnvarnes datoteka pri otvaranju setvaraju prethodne otvorene i si Pri rado se rasimm zonama treba biti vesma pažljiv i najvažnije, dosledan Treba strogo poštovati sledeća jednostavna pravila Prvo datoteku treba otvarati na početku procedure/modula n bnne je potrebna, a na kraju ztvarati Otvaranja svih potrebnih datoteka na početku programa nga preporučljivo zasm u slučaju kada ih ima jako malo ili ei stalno potrebna. Naime *MS-DOS* ograničava brnj istovremeno otvorenih datoteka (direktiva **FILES** u **CONFIG.SYS**) Ako pokušate se istovremenim otvaranjem svih datoteka dođi do da greške u izvršavanju programa. Treba izvršiti pažljivu analizu i držati stalno otvorena samo neophodne datoteke (Na primer- datoteku u kojo se upisuje dnevnik rada i slično) Pri otvaranju datoteke radnaj zoni obavezno treba dodeliti neko prepoznatljivo ime i nadalje datotom u toj zoni pristupati isključivo pomoću tog imena. Otvaranje treba uvek vršiti u slobodnoj radnoj zoni kako se ne bi prekkom setvorile neke potrebna datoteka. Za to se koristi opcija **NEW** pri otvaranju datoteke: ova opcije nalaže otvaranje datoteke u prvoj slobodnoj radnoj zoni Na primer

```
SELECT 7
USE Ucenici NEW
USE Ocenici NEW
```

Naredba **SELECT** služi za izbor jedne od radnih zona. Kada je moževa jedna radna zona, može se pristupati podatni ma u datotem koje je otvorena u tej zoni Na primer ako datoteka radnice ima polja **IME** i **PREZIME**, a datoteka **PLATE** polja **BR_RADNIKA** i **IZNOS** zanle naredba **SELECT Radnici** može se napisati ? **IME, PREZIME** dak su polja **BR_RADNIKA** i **IZNOS** nedefinisana. Ako se uvek naredba **SELECT Plate IME** i **PREZIME** postaju nedefinisani o nnde se prisiupati polnme **BR_RADNIKA** i **IZNOS**. Take polje mogu osim iste imena u raiičitim različitih datoteka. polim je uvak aktivna jedna radna zona. vadijva mi polja iz te zone

Često je vrlo česte potrebne koristiti podatke u nekoliko različitih zona (i unutar jednog zona, na primer) omogućen je pristup poljima iz drugih oblasti pomoću samog primenom operatora **->** Ovaj operator služi se prvenstveno (u slučaju jed nog izraza) promenu radna zona. Na primer

```
? Ime, Prezime, Plata
? Ucenici->Ime, Ucenici->Prezime, Plate->Iznos
```

Ovde se ništna vrednost polja **BODOVI** iz radna zona **PLATA** i polja **IZNOS** zone **NAGRADE**. Ako se ne navede alias, ima se polje iz tekuće radna zona. Ako takvo ne postoji, traži se mezarajuća promenljiva se tim nnzenom, o ako ni ona ne postoji, prijavljuje se *run-time* greška Ovim prostiln vremenom polja, operator **->** podržava urrchunavanje i čitavih zemna, u ključnjuči i pnivovojna funkcije:

```
? Ucenici->(Ime+" "+Prezime)
```

U ovom primeru se proveravn da li u radnoj zoni u kojoj je otvorena datoteka učenici pohnnvač dužna da izraza datoteke I zna broj se računa obavezno treba zatvoriti u zagrade

OPERACIJE SA ELOGOVIMA I POLJIMA

Kada se **DBF** datoteka kreira funkcijom **DBCreate**, u njoj se postaji nijedan slog Program koji upravlja bezma podatnka treba da omogući upis novih slogova. kao i izmene postojećih. Dodavanje novog sloga se izvodi tako što se prvo ngnđe prazan slog sa kraj datoteke a zatim se popnnu željenom podacima. Naredba **APPEND BLANK** dodaje prazan slog sa kraj aktivnu datoteku Pakazivač tekućeg sloga se tada automatsku pomera na dodati slog Upis podataka u slog vrši se naredbom **REPLACE**. Sledeći primer pokazuje kako se dodaje novi slog u nekoj datoteku.

```
SELECT Ucenici
APPEND BLANK
REPLACE u_Ime WITH "Žika"
```

Za izmenu vrednosti u slogu takode se koristi naredba **REPLACE** ona željeno polje postavlja na evu vrednost Tako se n prethodnom primeru u polje **u_Ime** datoteke **UCENICI** u pnnja vrednost „Žika".

Umesto naredbe **REPLACE** potpuno ravnopravno se može koristiti naredba za dodeljivanje = unnsta prethodnog se može pisati **: u_Ime := "Žika"** Pri tome vaša neka ograničenja ako se naredba dodeljivanja napiše ovako *Clipper-ov recitime* sistem ne može znati da li se ovi upis u polje **o_Ime** ili u memorijsku promenljivu se istim nazivom Zato se u svakvim sinčajevima predmet dnje memorijskom promenljivim pa će prethodne naredba kreirati **memorijsku** promenljivu **u_Ime** dak će vrednost polje ostati nezmenjena. Ovo se može izmeniti na dva načina prvje da se navede alias radne oblasti u kojoj se nalazi polje (ili ključna reč **FIELD**) Na primer

```
USERTEC->( DBS() ) := "Zlika"
FIELD->( DBS() ) := "Zlika"
```

Drugi način je da se u teks deklaraše kao polje, na šta je jedna šema deklaracija **FIELD**:

```
FIELD Ime, Prezime, Godrod, Adresa, Telefon
```

Posle ovoga će se pri svakom referisanju na u teksto podrazumevati da se radi o polju u radnoj zoni se alijasom **Usenter**.

I pored postojanja drugih mogućnosti preporučuje se da se se dodeljivanje vrednosti koriste naredbe **REPLACE**. Na slog nije vezana ni funkcionalnost (i naredba **REPLACE** se preimenovanu u ponovo naredbu dodeljivanja :=) već u dizajnerski programa. Naime, naredba poput **REPLACE** je daleko ugodnije od naredbi dodeljivanja, pa je znatno lakše individualizovati mesto koje pristupaju teko podataka, što je ponekad potrebno.

Pre uzmene vrednosti polja u slogu potrebno je pomeriti pokazivač sloga na slog koji želimo da izmenimo na primer naredbom **GO**. Na primer ako želimo da izmenimo slog broj 20 pre odgovarajuće naredbe **REPLACE** izvršićemo **GO 20** ili neku drugu naredbu koja će pomeriti pokazivač na odgovarajuće mesto

Brisanje nepotrebnih slogova u datoteke je nešto složenija operacija. Tu se razlikuju fizičko i logičko brisanje sloga. Kada se slog logički izbriše on se ne uklanja iz datoteke već se samo markera specijalnim internim markerom. Ovo markiranje se obavlja naredbom **DELETE** a skidanje markera sa brisanje obavlje naredba **RECALL**.

Kada treba slog fizički izbrisati iz datoteke, mora se cela datoteka prepakovati, pri čemu se svi slogovi markirani na bri sanje zaista i brišu. Ova operacija n zavisnosti od vaišnog **DBF** datoteke može prilično potrajati. Zato je preporučljivo da se brisanje sloga odvoja u dva faze. Prve se samo svi slogovi koje treba izbrisati naredbom **DELETE**. Posle toga se poziva naredba **PACK** koja fizičku brše sve označene slogove iz datoteke. Ako se u međuvremenu predomišlite se slogova označenih za brisanje možete „skinuti" oznaku na brisanje naredbom **RECALL**. No primer ako treba izbrisati slogove broj 10 120 i 500

```
GO 10 ; DELETE
GO 120 ; DELETE
GO 500 , DELETE
PACK
```

Funkcija **Deleted()** vraća logičku vrednost koja označava da li je tekući slog označen za brisanje. Na primer

```
DELETE
? Deleted() // FALSE
RECALL
? Deleted() // FALSE
```

Ovaj mehanizam vam omogućava da koristite potpuno suvišnog čekanja pri svakom brisanju sloga obaveštenje slogova za brisanje je trenutna operacija, pa takam slog programa možete omogućiti slobodne označavanje nepotrebnih slogova i fizičko brisanje se može uraditi tek na kraju programa ili ak spisteme naredbom korisnika. Ova se pokazuje kao vesma pogodno jer ostavlja korisniku mogućnost da se predomisli i „vrati" neki slog koji je grešnom izbrisan – naravno, pre uvođenja naredba **PACK**. Postoji i naredba **ZAP** koja fizički briše sve slogova u datoteci.

Pri radu ne datotekom nekad je potrebno saznavanje logičkog izbrisanog sloga: dak je ponekad važno da ni vidljivi svi slogovi uključujući i izbrisane (na primer, korisnik treba da ekine oznaku „obrisano" se logički izbrisanih slogova). To se u *Clipper*-u može podešavati naredbom **SET DELETED**. Ako se izvrši naredba **SET DELETED ON**, izbrisana slogovi će postati „nevidljivi" sa obrazi **SET DELETED OFF** sušte ravnopravan tretman za sve slogove kao obična sta.

PRETRAŽIVANJE BAZE PODATAKA

Smisao baze podataka se sastoji u tome da se na jednom mestu čuva velika količina podataka i da se na zahtev dobijaju traženi. Vesma važno mesto u Clipper-u zauzimaju naredbe za pretraživanje baze podataka. Od izuzetka kada korisnik baze podataka zatraži neka podatak do trenutka ispisa rezultata na ekran vrede mnogi postavi traženje i nalvažnija potrebnih podataka može se izvesti na mnogo različih načina. Koji način će te biti mora odlučiti programer pri pisanju programa. Clipper podržava dva osnovna načina pretraživanja datoteka baze podataka. sekvencijalno traženje po izrazi i traženje pomoću indeksa.

Traženje po izrazi se obično obavlja kada nije unapred (za vreme pisanje programa) poznat tačan izraz po kome se traži slog. Takvo traženje obavlja naredba **LOCATE**. Ova naredba ima dva parametra: prvi je izraz koji opisuje slog koji želimo da pronađemo, a drugi je opcionalni parametar **WHILE** koji zadovoljava neki uslov u programu ukoliko ispuni ovaka

```
LOCATE FOR Ime = "Petar" .AND. Grad = "Beograd"
IF Found()
   // Slog je pronađen
ELSE
   // Ne postoji slog za zadati uslov
END IF
```

Vrlo često postoji više slogova koji zadovoljavaju zadati u slov Naredba **LOCATE** pronalazi prvi slog koji zadovoljava uslov počev od prvog sloga u datoteci. Ako je potrebno naći i mesto sloga, koristi se naredba **CONTINUE**. Ona nastavlja započeto traganje počev od tekućeg sloga, pod uslom uslovima kao i naredba **LOCATE**. No primer

```
LOCATE FOR Brlk = W_Brlk
IF Found()
   ? Brlk
   CONTINUE
   IF Found()
      ? "Postoji i drugi slog
         koji zadovoljava uslov ", Brlk
   ELSE
      ? "Nema drugog sloga!"
   END IF
ELSE
   ? "Nema sloga koji zadovoljava uslove!"
END IF
```

Veliku je sredba ako se ne prethodni operaciji naredba **LOCATE** koristi u ukluu – ova naredba uvek traži slog od početka datoteke po će uvek „naći prvi slog koji zadovoljava uslov U svakvom slučajuvima samo prvi slog treba tražiti naredbom **LOCATE** a sve ostale naredbom **CONTINUE**.

Ovim tražnja slogo koji zadovoljava zadati uslov mogu se izdvojiti svi slogovi koji zadovoljavaju uslov Za to služi naredba **SET FILTER** Korišćenjem te naredbe zadaje se logički u slov koji u aktivnoj datoteci prvobno izdvaja samo ona slogove koji zadovoljavaju zadati uslov Važno je znati da je ovo suštvarje samo privid ne da se datoteke „smanjuje" ngrestanim računanjima logičkog izraza za svaki slog. Ova operacija može biti vesma osetljiva u vše razloga Osnovni razlog je što se pri svim kasnijim operacijama računa isnova logički izraz za sva ki slog kome treba pristupiti što može biti vesma sporo Zato upotreba ove naredbe treba što je moguće više ograničiti Ui sledeem primeru je prikazano izdvajanje svih slogova u datoteci **FARTURE** koji sadrža podatke starije od nekog datuma.

```
SET FILTER TO F_Dat <= Date() - 30
// Filter će aktivirati poberanje
// potrživeća na poštenu datoteke
GO TOP
```

Filter se isključuje ili specifiranjem nevog filtera naredbom **SET FILTER** ili naredbe **SET FILTER TO** bez argumenata, koja ukida aktivan filter

Kada se postavi filter na slogova jedna datoteke ova operacija se datoteke radu na osnovu uobičajen načna, a tim što slogovi koji na ispunjavaju zadat uslov postaju „nevidljivi" Naime kada pokazivač sloga nađe na takav slog on ga jednostavno preskače i pomrve se na prvi sledeći koji zadovoljava uslov Filter u jednoj radnoj oblasti ostaje aktivan i posle završavanja drugog rada oblasti ovaka radna oblast vesmora ima svoj filter : on se može ukinuti samo naredbom naredbom **SET FILTER** ili zatvaranjem datoteke naredbom **CLOSE**

Izraz koji se zadaje a naredbi **SET FILTER** može biti proizvoljan *Clipper* izraz. Dodatno može se upotrebiti i makro operator kako bi se postavio upit po izrazu koji nije unapred znan, nego ga, na primer zadaje korisnik

```
ACCEPT "Unesi uslov: " TO cExpr
SET FILTER TO &cExpr
```

INDEKSIRANJE

Naredba **LOCATE** realizuje takozvano sekvencijalno pretraživanje to je naziv se jednostavnu provere izraza za svaki slog u datoteci. Naravno ave operacije u slučaju velih datoteka može ostati potrajati. Ukoliko se upitanje u datoteku indeksa sekvencijalno pretraživanje na slog na koji se izraz odnosi indeksa se prikazuje upravanja organizuju u specijalnu strukturu koja se zove **B-stablo**. Sada pretraživanje ide na sledeći način u mestu da se redom ispituju slogovi u osnovnoj datoteci pretražuje se datoteka indeksa Ovo pretraživanje ide daleko brže od pretraživanja sekvence datoteke sbog strukture postoje algoritmi koji omogućavaju da traženo ibre nađete podatake u B-stablu Kada se nađe traženi indeks pomoću pokazivača se dobijaju pravi podaci u osnovnoj datoteci

Da bi se kreirali indeksi n *Clipper*-u prve je potrebno kreirati indeksnu datoteku. Ona se kreira naredbom **INDEX**.

```
USE Ucenici
INDEX ON Upper(U_Ime) TO Ucn_Ime
```

Rezultat izvršenje ovih naredbi je da se kreira indeksna datoteka **UCE_IME.NTX** koja sadrži ave vrednosti izraza Upper(u_Ime) za svaki slog datoteke **UCENICI**. Ova operacija se vaše može obaviti samo potrajati, pa je dobro okarvešteavati korisnika o tome šta se događa. Naredba INDEX zete podrkova opciju **EVAL** koja omogućava da se pri obradi svakog sloga izvrši određena funkcija.

```
INDEX ON u_Ime TO Ucn_Ime EVAL Proces()

FUNC Proces
   cOut("record " + ;
      Str(RecNo() + "/" + Str(LastRec())))
RETURN True
```

U navedenom primeru se pri obradi svakog sloga (odnosno pri uplsu vrednosti kljuha u indeksnom datoteku) izvršava funkcuja **Proces** koja ispisuje podatke o broju sloga koji se obrađuje Ova funkcija **obavezno** mora da vrati vrednost logičkog tipa Indeksiranje se vrši sve dok ova funkcija vraća vrednost **.T.**.

Ukoliko datoteka ima dosta slogova, izvršavanje svakve funkcije na svaki slog može vesma usporiti indeksiranje, pa sete postoji opcija **EVERY** koja specifcire na kolko n obrađenih slogova treba izvršiti funkciju **Proces**: **EVERY 10** će ak izvršiti funkciju na svakih 10 obrađenih slogova

Vesma važna opcija pri indeksiranju je : **ASCENDING** ona određuje da se indeksi slažu po rastućim vrednostima. Suprotan efekat ima opcija **DESCENDING** koja indekse slaže po opadajućim vrednostima.

Posle kreiranja jedne ili više indeksnih datoteka na trebe ih uvek imenu kreirali, već ne pri otvaranju datoteke navede ave ajene indeksne datoteke

```
USE Ucenici INDEX Ucn_Ime, Ucn_Mat
```

Umesto ovaga može se pisati i.

```
USE Ucenici
SET INDEX TO Ucn_Ime, Ucn_Mat
```

Svaka izmena osnovne datoteke automatski menja i sve pridružene indeksne datoteke ali samo ako su otvorene. To znači da uvek pri otvaranju datoteke treba otvoriti i sve njene indekse, jer se u suprotnom. nastale izmene može videti na svim indeksnim datotekama.

I konačno, kako se kreirate redosled, ako datoteka ima otvoren i aktivan indeks se pretraživanja se izvesne naredbe **LOCATE** može koristiti naredba **SEEK** – ova naredba traži zadatu vrednost u aktivnom indeksu. Na primer traženje učenika po imenom "Pera Perić" može se izvesti naredbom **SEEK "PERA PERIC"** Slično kao na naredbu **LOCATE**, ako slog se ime imenom postoji pokazivač sloga će biti na tom slogu, a funkcija **Found()** imaće vrednost **.T.** Pri pretraživanju je neophodno navesti tačan izraz u polju. Zato se koriste funkcije kao što je **Upper** za konverziju u velika slova : ul

Za svaku **DBF** datoteku može se kreirati više indeksa i po potrebi aktivirati jedan od njih. Ujednačen trenutku može biti aktivan samo jedan indeks. Koji će indeks biti aktivan određuje se naredbom **SET ORDER**. U sledećem primeru se kreiraju tri indeksne datoteke.

```
USE Radnici
INDEX ON R_Matbr TO Radnici1
INDEX ON R_Ime TO Radnici2
INDEX ON R_Staz TO Radnici3
```

Posle uvršćenja svih naredbi, datoteke **RADNICI** se može otvoriti na sledeći način

```
USE Radnici INDEX Radnici1, Radnici2, Radnici3
```

Posle ovoga je aktivan šedeo iz datoteke **RADNICI1**, a to je **R_MATBR**. Bez obzira što prestala dva indeksa nisu trenutno aktivna, odgovarajuće **NTX** datoteke će biti korektno ažurirane se svakom promenanm **DBF** datoteka. Ako se svrši naredba **SET ORDER TO 2** aktivirade se indeks iz datoteke **RADNICI2**, opšte **SET ORDER TO N** aktivire N-ti indeks **SET ORDER TO 0** isključuje sve indekse, ali indeksne datoteke ostaju otvorene, pa će se sve izmene na DBF datoteci normalno prenositi i na indekse.

Održavanje indeksnih datoteke na osnovu redosled broja prilikom otvaranja može biti vesma neugodno. sledeća promena redosleda otvaranja indeksnih datoteka može potpuno promeniti smisao programa! Zato je mogućnost da se svakom indeksu dodeli simbolično ime (tag) koje služi za identifikaciju. Da bi se jednom indeksu dodelilo simbolično ime, koristi se opcija **TAG** pri kreiranju indeksa.

```
INDEX ON R_Ime TO Radnici2 TAG Ime
```

Sada se umesto **SET ORDER TO 2** može pisati **SET ORDER TO TAG Ime** što je daleko preglednije

U svaku **NTX** datoteku može se upisati samo jedan indeks. To znači da je potrebno uvek kod otvaranja **DBF** datoteka otvarati i sve pridružene **NTX** datoteke. Pošto je maksimalni broj otvorenih datoteka u *MS-DOS*-u ograničen, ovo ponekad može biti neprijatno. Na primer ako vam je i stavremenu potrebna jedna **DBF** datoteka i ako svaka ima po tri indeksne datoteke bilo istovremeno otvorena najmanje 20 datoteka Neki drajveri za pristup podacima omogućavaju da se u jednu datoteku smesti više indeksa takvi su **DBFCDX** i **DBFMDX** drajveri koji se dobijaju na *Clipper* 5 2 Prvi kreira **CDX** indekse kompatibilne sa *FoxPro* programom a drugi **MDX** indekse kompatibilne sa *dBase IV* Kad se ovakvim indeksa je vesma sličan kao kad je aktivaran **DBFNTX** drajver s tim što postaje neke dodatne mogućnosti Ne primer može se napisati

```
USE RADNICI.CDX,"IME"
INDEX ON M_MATUR TO Maturski
INDEX ON M_Ime TO Radnici
INDEX ON M_Staz TO Radnici
```

Sada će svi indeksi biti smešteni u datoteku **RADNICI.CDX**, pa će biti smanjen broj otvorenih datoteka.

Kao teraz se indeksiranje može se upotrebiti prosevuljtas i xraz tipe string brej datum i logički tip. Prilikom indeksara iga se za svaki slog računa vrednost izraza i ta vrednost se zajedno se pokazivačem na slog spasuje u indeksnu datoteku. Indeksiranjem slogovi privodno manjaju redosled. pada nada na prvom mestu bio slog brej jedan. već slog koji ima vrednost indeksa koje je najmanje (ako je izabrano indeksiranje se opsi sam ASCENDING). Na primer ako su sortirana neka imena, prvi slog će se odnositi na osobe čije ime počinju na "A" i sl. Za svaj redosled se često pada da je logički Naredbe koje pomeraju pokazivač slogu pošiaju ovaj redosled. Na primer ako je tabela sortirana po imenima, naredbe **GO TOP** izaziva sluk na slog se pokazina na osobu čije ime počinje slovom „A" bez obzira što taj slog može biti i poslednji u tabeli. Naredba **SKIP** takođe poštuje ovaj redosled Međutim funkcija **Recno()** vraća fizički brej slog bez obzira što on logički može biti i na prvom mestu. Take indeks na izvestan način i sortira datoteku. sve naredbe „vide" **DBF** datoteku kao da je sortirana po kreatiini aktivnom indeksu. Na primer. ako u datoteku **PROBA** koja ima numeričko polje **P1** upišeme slogove 2, 3 5 4 1 tim redom, naredbe za pregled datoteka ispisuje slogove onim redosledom kaka su i uneti Ako se datoteka indeksira po polju **P1** datoteka će se ponašati kao da je slog 1 prvi upisan, zlog 5rojevi de biti upisanu u sledećem redosledu: 2 3 5 4 1. Važno je napomeni da indeksna nas fizičku menja na, take da je brej slogova sačuvan. Na sledeću tabeli vide se rezultati.

PRE INDEKSIRANJA	
VREDNOST	RECNO()
22	1
33	2
55	3
44	4
11	5

POSLE INDEKSIRANJA	
VRSONOST	RECNO()
11	5
22	1
33	2
44	4
55	3

Indeksiranje po datumskom izrazu je dosta časta, all da bi se izvršile korektne sortiranje takvih podataka (datumi se poređani po hronološkom redosledu) potrebna je indeksirati korišćenjem funkcija **Dtos()** koje datum transformiše u string pogodan za indeksiranje.

U prethodnim primerima uvek je indeksirana sela **DBF** datoteke Međutim, pri indeksiranju se mogu navesti i opcije koji određuju skup slogova koji će uči u indeksnu datoteku. Na primer

```
INDEX ON Upper(Ime) TO Imena FOR Poen1 >= 100
```

kreira indeksam datoteku **IMENA** koji sadrži indekste samo se slogove za koje vaši izraz **Poen1 >= 100**. Kada se aktivira ovaj indeks bilo vidljivi samo slogovi koji zadovoljavaju ovaj uslov. Na ovaj način se mogu izdvajati podaci iz **DBF** datoteke na daleko efikasniji način od naredbe **SET FILTER**. Jedina mana ovog metoda je što prethodno treba kreirati indeksnu datoteku što može potrajati u slučaju velog broja slogova. Ka da se kasnije dodaju slogovi u datoteku indeksna datoteke se **FOR** uslovma i zlaje „filtrira" slogove za svaki novi slog se računa indeksa i uključuje u indeksnu datoteku samo ako zadovoljava zadati uslov

Sliđeu efekat ima opcija **WHILE**. Ona služi se još preveniji sužje skupe slogova koji će uči u indeks. kada se zada **FOR** uslov obrađuju se *svi* slogovi u aktivnoj datoteci Za razliku od toga, kada se pada uslov **WHILE** obrađuju se slogovi počev od tekućeg sve dok je uslov ispunjen To znači da će ista obrađeni samo jedan deo slogova u bazi podataka.

```
GO 500
INDEX ON Broj TO MOJ_INDEX FOR ...
```

Ovaiiuv indeks je privremen, tj. pri kasnijem dodavanju slogova u datoteku neće biti ponovo uzimam u obzir

Sličan efekat ima opcija **NEXT** ona određuje obrazu specifičnaroog broja slogova počev od tekućeg

```
GO 200
INDEX ON ... TO MOJ_INDEX NEXT 500
```

Ukoliko obrazu treba ograničiti na deo od tekućeg sloga da kroja datoteka koristi se opcija **REST**

OPERACIJE SA DATOTEKAMA

Operacija se **DBF** datotekama se svode na kreiranje novih datoteka, upravnje novih izmenu i brisanje postojećih slogova. Osim toga, **DBF** datoteke se mogu prestrukturi mogu se jedvojiti podaci iz njih i slično Neke od naredbi koje omogu davaju ove operacije već se spomenute na primer **APPEND BLANK** dodaje novi slog u **DBF** datoteku, **REPLACE** služi na izmenu postojećeg sloga, itd. Neke od ovih naredbi mogu se primenjivati na vrše slogova odjednom. Na primer ako je potrebno polje **Broj** postaviti na 0 n svim slogovima n datotesi koristi se opcija **ALL** naredbe **REPLACE**:

```
REPLACE Broj WITH 0 FOR Broj<>0
```

Sličre se može zadati uslov na izdvajanje slogova koji će biti obrađeni

```
REPLACE Broj WITH 5 FOR Broj<>0
```

Bada će biti obrađeni svi slogovi kuji imaju polje **Broj** nauve od nula. Osim **FOR** uslova, može se koristiti i uslov **WHILE** kuji zadaje skup slogova počev od tekućeg sloga, ave dok je se dati uslov ispunjen

```
REPLACE Broj WITH 0 WHILE Poen1()>=100
```

Opcije **ALL REST NEXT** i **RECORD** služe se određivanje skupa slogova koji će biti obrađen i koriste se u skoro svim naredbama na obradu vše slogova u datoteci. **ALL** označava sve slogove u datoteci, **REST** sve slogove od tekućeg da kraja, **NEXT <n>** naredbih **<n>** slogova i **RECORD <n>** slog broj **<n>**

Za listanje podataka se koristi naredba **LIST**. Ona dodata na izraza ispisuje na ekranu. Pri tome se mogu koristiti sve raknrgane opcija na izbor skupa slogova.

```
LIST Ime, Prezime ALL FOR Grad = "Beograd"
```

Ova naredba ima isto dejstvo kao i

```
WHILE !EOF ()
  IF Grad = "Beograd"
    ? Ime, Prezime
  END IF
  SKIP
END DO
```

Postoji štav niz naredbi koje obavljaju razne operacije se slogovima aktivne datoteke koje rade po sličnim principima kao i opisane naredbe. Na primer naredba **SUM** za sabiranje izraza na nivou cele datoteke, **COUNT** za prebrojavanje slogova koji zadovoljavaju određen uslov **AVERAGE** za računanje prosečne vrednosti izraza, itd. Od važnih treba pomenuti da i vaš opisane naredbe **DELETE** i **RECALL** se brisanje slogova podržavaju opcije sa „oklast" valzaja. Take se može pisati.

```
DELETE ALL FOR Grad="Beograd"
```

Sve sve naredbe su realizovane prako vsema berisne šun kaja **DBEval**. Ona omogućava izvršavanje bodnog bloka na zadate skup slogova u datoteci Ako se navede samo kodni blok, ova funkcija se svaki slog datoteka izvršave zadati kodni blok. Na primer ako dabta da odštampate polje **Plata** iz svakog sloga tapeda datoteke, izvršava se sledeća naredba.

```
DBEVAL( {|| QOut(Plata) } )
```

Ova funkcija dozvoljava i opcana zadavanje uslove kajcee se postuje isti efekat kao i kljućima reima **ALL**, **REST** itn. Pri na mtuksa ove funkcije je

```
DBEval( <bKod> [,
  [<bFOR>] [,
  [<bWHILE>] [,
  [<nNext>] [,
  [<nRecord>] ] ] ] )
```

Jedini obvezan argument je kodni blok kaji treba izvršati deb su svi ostali opcioni. Ne primer, prethodna naredba koja nji naja broj 0 u sva slogove kaji zadovoljavaju uslov može se napi sati.

```
DBEVAL( {|| Broj:=0 } , {|| Broj<>0 } )
```

Sva standardne naredbe na već se bosma podataka se prepreosnirnju u pavine funkcije **DBEval**. Keko izgledaju te definisi je mašinski videte u datoteci **STD.CH** Korišćenjem ove vrste korisne funkcije mogu se lako dodavati nepstvene naredbe es već se bazom podataka koje će podržavati sve opcija za zadavanje opsega slogova **FOR, WHILE** i sl

RELACIJE IZMEĐU DATOTEKA

Kada se kreira struktuje jedna **DBF** datoteke, uvek treba znati jedno polje koje je različite se svaki slog datoteke. To je je **ključ** datoteke i na osnovu njega se bosnije traži određenu podatak. Na primer ako se pravi tabela koja će sadržati podatke o studentima obevezno treba uvesti jedno polje koje se razlikuje od ostalih. Ima i prašine nisu pogodan izbor- ioje taka rečne da se nadu dve osobe koje imaju iste ime i prezime. Nolte bolji izbor je matični broj ili broj dosijea U slučaju da na postoji podatak koji je jedinstvenu se svaki slog može se u vesti podatak koji će program sntomatski garantisati. Osim u slova jedinstvenosti manipulacije se ključem je mnogo jednostavnija nego se nekim podatkom tipa „Petar Petrović" Eva kako neke izgledati datoteka **STUDENTI**

IME	DOSIJE	ŠIFRA
Pera Perić	346/00	1002
Mika Mikić	120/91	1003
Zora Zorić	340/90	1004

Kada je u nekoj drugaj datoteki baza podataka potrebun referisanje na određeni slog datoteke studenata (na primer datoteki položenih ispita) umesto imena i prezimena koristi se ključ odgovarajućeg sloga.

ISPIT	DATUM	STUDŠIF
Matematika	12.03.91	1003
Biologija	11.04.91	1002
Fizika	18.04.91	1003

Sada je na snimu šifre studenta take određuje na kog se studenta podaci odnose Pretpostavimo da je datoteka studenata indeksirana po šifri. Tada će sledeći programa ispisati sva početke o položenim ispitima.

```
USE Studenti INDEX Studenti NEW
USE Ispiti NEW
WHILE !Eof()
  SELECT Studenti
  SEEK Ispiti->Studšif
  SELECT Ispiti
  ? Ispit, Ocena, Br
  SKIP
END DO
```

Umesto šifre studente poda se nalaz i ispisuje odgovarajuće ime iz datoteke studenata.

Za svakve operacije postoji naredba **SET RELATION** kaja omogućuje automatsko traženje podataka. Za prethodni primer treba pisati.

```
USE Studenti INDEX Studenti NEW
USE Ispiti NEW
SET RELATION TO Studšif INTO Ispiti
WHILE !Eof()
  ? Ispit, Ocena, Br
  SKIP
END DO
```

Sada na treba menjati ručnu xenu pretraživati drugu datoteku ol. dovoljna je naveden nuzaj naredbe **SET RELATION**. Dejstvo ove naredbe je dosta jednostavno uvek kada se pomeri pokazivač sloga u datotesi iz koje je postavljena relacija, izvršava se naredbe **SEEK** n datoteci na koju je relacija „u nosrena". Kada se drge datoteke iz koje se postavlja relacija **redišelj** u datoteka na koju se postavlja relacija dete.

Broj relacija po jednoj radnoj zoni na mora biti jedan. iz jedne radne zone se može postavita relacija na više datoteka. Na primer datoteku ispite se povezuje na datotekum studenata i datotekum profesora.

```
SET RELATION TO Studšif INTO Studenti,
   TO Profesor INTO Profesori
```

Pri tome je datoteka profesore organizovana na slično način ime i datotekm studenata u osnovnoj datoteci su svi zadati i šifra, a u datoteci ispita se nalaze šifre odgovarajućih profesora. Sada je dajstva naredbe prokreno i na radnu sonu **PROFESOR**. kada se pomeri pokazivač u osnovnoj zoni, pomeraju se pokazivači slogova i u zoni **STUDENTI** i u zoni **PREDMET**.

Dajstvo naredbe **SET RELATION** ukida sledeća postavljanje relacije ili navođenje **SET RELATION TO** bez argumenata. Ponekad je potrebno na postojeće relacije dodati nekn novu. Toda se koristi opcija **ADDITIVE**.

```
SET RELATION TO Studšif INTO Studenti
SET RELATION TO Profesor INTO Profesori ADDITIVE
```

Radna zona na kaju je postavljena relacija (date) nosđe takode biti spojena se nekom radnom zonom naredbom **SET RELATION**. Tada će pomeranje pokazivača sloga u osnovnoj zoni prouzrokovati pomeranje pokazivača u svim sonama se kaji ose je uspostavljena relacija a te dalje prouzrokuje pomeranje pokazivača u svim sonama se kojima su povezane ove zone itd. Take se mogu ostvariti višestruka veza između više datoteka. Uslov koji se pri tome mora ispuniti je da se ne napravi ciklus radne sona A je povezana se zonom B, B je povezana saC a ova on A.

VIŠEKORISNIČKI PROGRAMI

Clipper podržava sredu programa koji će raditi u mreži ili u nekom višekorisničkom okruženju. Pakte sva funkcije i na radna na sredu rade na vesma jednostavnom nivou. *Clipper* programi se bez problema mogu izvršavati na skoro svim postojim mrežnim operativnim sistemima

PRINCIPI VIŠEKORISNIČKOG KADA

Mrežu skitao dni vide računara kaji su na neki načnu spojeni i mogu međusobno da šela resurse: datoteka, štampače itd. Osnovni problem u višekorisničkom vrši su podaci kojima istovremeno može pristupiti više korisnike Uloga „mrežnih"

naredbi je da sinhronizuju rad korisnika nad zajedničkim podacima koji se čuvaju u **DBF** datotekama. Šta te znači? Neka, na primer, jedan korisnik otvori datoteku naredbom **USE** Postavlja se pitanje šta će se desiti ako neki drugi korisnik takođe pokuša da otvori istu datoteku. Ne prvi pogled nema ničeg ružnog da mu se to ne dozvoli. I zaista, ukoliko bi jedan od korisnika samo štitio podatke uz postoji razlog da se zabrani pristup ostalima. Međutim, Re ako oba korisnika počnu istovremeno da menjaju podatke u datoteci naredbom **REPLACE**? Ili još gore: oba menjaju iste slogove? Ukoliko bi to akcija bile vremenski unkrsnaisovane sve bi bilo u redu. Međutim, može se desiti da se vreme izmene nekog polja drug korisnik promeni vrednost nekog već izmenjenog polja i tako naruši ispravnost podataka. Na primer dva korisnika pokuša vođi da upišu kadatke s dve osobe

KORISNIK A	KORISNIK B
1) Ime = "Pera"	Mesto = "Sombor"
2) Prezime = "Peric"	Ime = "Mika"
3) Mesto = "Beograd"	Prezime = "Mikic"

Korisnik A radimo upisuje podatke o nekoj prezimenu i adresu ali korisnik B „kvari" te podatke istovremenom upisom (naravno te se može pomnožiti i obrnuto: i korisnik A kvari podatke korisniku B). Tako dalazi do konflikta, zadatak programa je da kontroliše pristup podacima i da obezbedi da nijedan korisnik ne može oštetiti podatke.

OTVARANJE DATOTEKA

Programi predviđeni za rad u mreži se na različiti previ će od jednokorisničkih programa po naredbama koje se koriste: vešina naredbi radi potpuno ista. Jedina razlika je u tome što neke naredba ne mogu uspeti ili mogu se desiti da se izvrši naredba, a da se sedata operacija opak ne izvrši. Na primer ako se na jednokorisničkoj mašini izvrši naredba **USE Test**, ona će uvek uspeti (naravno pretpostavlja se da postoji datoteka **TEST.DBF**). U mreži je atvar malo izmenjena naredba **USE** ne mora uspeti da otvori datoteku. Razlog može biti da je drugi korisnik tu datoteku otvorio za ekskluzivno korišćenje. Zato se posle naredbe **USE** obavezno vrši provera ishoda operacije. Za to služi funkcija **NetErr**. Ova vraća logičku vrednost koja označava status prethodne operacije. Znači, otvaranje datoteke u mreži treba izvršiti ovako:

```
USE Test
IF NetErr()
  ? "GREŠKA"
  RETURN
ELSE
  // obrada
END IF
```

Šta će biti konkretne urađeno u slučaju neuspeha, stvar je i ubera programera. Takuću proceduru svakako ne treba zaniavi ti per podaci iz datoteke **TEST** neće biti dostupni. Najbolje je koje može pauze posavriti postupak ili prepustiti korisniku da odluči. Kako se to radi možete videti u datoteci **LOCKS.PRG** koje se dobije uz *Clipper*

U prikazom izvršavanja naredbe **APPEND BLANK** treba proveriti status nije dnevnjene da se u jednu datoteku istovremeno dodaju dva sloga, pa program jedino od korisnika može uspeti sa dodavanjem. Zato i posle ove naredbe obavezno treba koristiti **NetErr** se proveru statusa operacije.

Pri radu sa datotekama u mreži postoje razni nadzovi koji određuju prave svakog od korisnika kaj koriste najpažnije datoteke. Najstrožiji je **ekskluzivni** način u kome korisnik koji otvori datoteku jedini koristi (upisuje i čita) podatke Kada se datoteka ekskluzivno otvori ujedan drugi korisnik je on može otvoriti sve dok je prvi ne zatvori. Pošto taka podacima pristupa samo jedan korisnik, nema nikakvih problema oko pri stupa i programu može raditi kao u jednokorisničkom okruženju. Datoteke se otvara se ekskluzivne korišćenje opet jem **EXCLUSIVE**

```
USE Test EXCLUSIVE
```

Ovakva otvaranje će uspeti samo ako je datoteka potpuno „slobodna", tj. ako je nijedan korisnik ni na koji način pristupa nju otvorio

Program će biti najjednostavniji ako se datoteka uvek svako otvara. tada se treba pariti na akcije drugih korisnika prilikom obrade **Međutim**, ova otvaranja se ražunirve ruđu koristi i to ugiavnom kada treba uraditi neku ručnu operaciju kao što je indeksiranje datoteke Ukoliko želite da datoteku učinite dostupnom i drugim korisnicima, treba je otvarati se opcijom **SHARED**

```
USE Test SHARED
```

Ova opcija ima suprotno dejstvo od **EXCLUSIVE** i označava da datoteci posle otvaranja mogu pristupati i ostali korisnici. To znači da će i drugi korisnici moći da otvore već otvorenu datoteku iztim svakvim naredbom Ste se tiče čitanja podataka, to nema nikakvih problema pomenutim pokazivačima sloga, slogovi se učitavaju u memoriju odakle se normalno mogu koristiti.

OBRADA PODATAKA

Sve ostale obrade (izmena sloga, brisanje) neće uspeti bez prethodne pripreme pre operacija se mora sprečiti da o stali korisnici se pokvare integritet podataka. Tu se svodi na jednostavnu privremenu zaštitu pristupa ostalim korisnicima (tzv **zaključavanje**) Na primer, potrebno je izmeniti ime kadi sloga Ako se svrše naredba **REPLACE** bez prepuna program prekida rad se porukom **"Lock required"** Ste znači da je prethodno potrebno zaključavanje Za to se koristi fun kcija **DBRLock**.

```
IF DBRLock()
  REPLACE Broj WITH YZW
ELSE
  ? "POKUŠAJ KASNIJU INTERVENCIJU"
END IF
```

Vaša operacija će uspeti ako je sloga navedeni parametar) ili slog čija se broj navede. Vraća se status operacije. ako se zaključa T operacija je uspela i slog je zaključan o u suprotnom nije. Zaključavanje može uspeti ako je neko već zaključao slog. Kada je slog zaključan može se mnogati ali ta vaš samo se korisnik koji je izvršio zaključavanje ostali i dalje mogu dobiti zaključani slog ali ga ne mogu menjati.

Vaša opcionalna i funkcija **DBRLockList** koja vraća rednu brojeve slogova koji su zaključani Na primer

```
DBRLock(100)
DBRLock(200)
GO 300
DBRLock()
a := DBRLockList()          {100, 200, 300}
```

Ova se rekosi na slogove koje je zaključao jedan korisnik. ako je potrebno provjeriti da li je neki slog zaključan od strane drugog korisnika, treba pročeti zaključavanje naredbom **DBRLock**, pa se na samovi dobijeni status može videti da li je slog već zaključan.

U starijim verzijama *Clipper* o se sa zaključavanje pojedinačnih slogova koristila funkcija **RLock** koja je radila slično kao **DBRLock**, ali nije bilo moguće zaključati više od jednog sloga u radnoj zoni svaka sledeća zaključavanje bi automatski otključala prethodno zaključan slog Zbog ovog nedostatka je dalaka koja koristiti funkciju **DBRLock**.

Poneksd je potrebno uraditi neku operaciju koje menja veliki skup slogova u datoteci ponekad i sve slogove. Tada u mnogo jednostavnoj zaključavanja svakom sloga treba koristiti funkciju **FLock**. ona zaključava sve slogove u datoteci Ova kvo zaključavanje može uspeti ako je neko zaključano bar jedan od slogova. Funkcija **FLock** vraća status operacije ako je uspešno zaključavanje uspele vraća se vrednost **.T.** a **.F.** u suprotnom Kada se svako zaključa cela datoteka mogu se primanjivati naredbe koje menjaju skup slogova (**REPLACE ALL**, na primer)

Svi zaključani slogovi u aktivnoj radnoj zoni se otključavaju naredbom **UNLOCK**, tada ostali korisnici mogu da zaključavaju „oslobođene" slogove. Uključe se upotrebi opcija **ALL**, otključavaju se svi zaključani slogovi u svim radnim oblastima.

Postavlja se pitanje kada treba koristiti eksplicitnu otvaranje kada **DBRLock**, a kada **FLock** funkcija Ste se tiče ak sklusivnog otvaranje treba poštovati pravila da pri kritičnim operacijama ekskovzne treba datoteku otvarati ekskluzivna Takve operacije su **PACK**, **ZAP**, **INDEX** i sl U principu generalno treba izbegavati ekuzluživno otvaranje datoteke, jer su tada ostali korisnici koji žele da rade sa tim datotekama potpuno blokirani.

Ako je potrebna manjati veću grupu slogova u datotoci, koristi se **FLock**. Na primer takve su naredbe:

```
REPLACE Broj WITH 0 ALL FOR Broj <
DELETE ALL FOR Broj >= 8
```

Za uzmenu jednog sloga dovoljno je zaključati sloj funkcijom **DBRLock**.

Za operacije koje na menjaju već samo čitaju podatke iz datoteke u principu nije potrebno zaključavati datoteku. Međutim, ponekad je dobro i u takvim ahuljevima zaključati čitavu datoteku jer se time sprečavaju ostali korisnici da unesu nave podatke za vreme vaka operacije kaja analizira podatke u velkom broju slogova. Na primer potrebne je izračunati ukupnu vrednost platnog fonda en sve radnika u evidenciji To se jednostavno radi ovako

```
USE Plate SHARED
```

Međutim, šta ako neka druga korisnik u isto povrdava platu svih zaposlenih se 5% i u jo svrhu izvrševa sledeći niz naredbi.

```
IF DBRLock()
  REPLACE PLATE WITH PLATA*1.05 ALL
  UNLOCK
END IF
```

Može se desiti da prvi korisnik dobije zbe obrađenih i nekih neobrađenih slogova, što je nečelno pod korektnim rezultatom sumiranje će stari iznos ili iznos uvećan za 0 05 Zato u vakvim slučajevima treba pre naredbe **SUM** zaključati datoteku.

OBJEKTI U CLIPPER-U

Clipper ima u sebi elemente objektne orjentisanog programskog jezika pozitiji magućunć kreiranja objekata određenih klasa, slanja poruke svim objektima i sl Ove je vaoma važna osobina *Clipper* a jer omogućava znatno veću fleksibilnost prilikom izrade programe

OSNOVNO O OBJEKTIMA

Objnet je uov tip podataka uveden u *Clipper*-u 5 0 Jedan objekt je jedinstvenotvorene reduse, najčešći ulaz pr čemu svaki element ima svoje ime Objekti može sadržati kao elementa promenljive, ali i funkcije Promanljive koje pripadaju objektu se ponašaju slično kao i elementi niza mogu se menjati iz četaji odgovarajuma naredkama i rad se njima se ne razlikuje od rada sa običnim promenljivim Za pristup ijutarnim promenljivim objekta koristi se operator send ( ) Na primer ako objekat oTB ima internu promenljivu **freeze** taj promenljivoj se pristupe se **oTB:freeze** Ako je potrebno odkampati vrednost ove osterne promenljive, koristi se izdvajanje 7 **oTB:freeze**, kasnima vrednost ove promenljive takođe ida ne određenim načim:

```
aTB:freeze := 0
```

Rad sa internim funkcijama koje sadrži objekat se uoče radi kaje od svega. Funkcija može imati i neke argumente. Na primer

```
? oTB:getColumn(8)
oTB:forceStable()
```

Prva naredba aktivira internu funkciju **getColumn** objekta oTB i pri tome se proslednje broj kao argument. Razultat te funkcije se dodjuje promenljivoj **X**. Druga naredba aktivira internu funkciju **forceStable** objekta aTB i pri tom se zanemaruje povratna vrednost Interne funkcije objekata nazivaju se **metodi**.

U pravom objektno orijentisanom jeziku programar sam definiše strukturu objekata, tj njegove interne promenljive i funkcije To se radi u dva koraka prva se definiše tzv **klasa** objekata i to kako će se se svaku klasu definiše strukture (promenljive i funkcije)—a zatim se kreiraju objekti koji pripadaju tej klasi ij imaju tako definisane promenljive i funkcije. Na žalost, u *Clipper*-u se poda nije moguće kreirati sopstvene klase objekata, već postoje fetiri unapred definisane klase i programer jedino može da kreira i koristi objekte koji pripada ju im klasama. To su klase se ubtevanje vrednosti, klasa za obradu grešaka. klasa en tabelarni pregled datoteka i klasa za definisanje kalena tabelarnog pregleda

KLASA ZA OBRADU OREŠAKA

Klasa za obradu grešaka je najjednostavnije klasa. Objekti iz ove klase sadrže internih funkcija već samo interne promenljive Oni se obično ra kreiraju programski već ih **raz-ni***ma* sistem kreiru kada se dogodi neka greška u programu Tada se kreira jedne objekat iz ove klase i proslednje se kod nam bloku (odnosno funkciji) zaduženom za obradu grešaka. Ako se ne uzmesi, ovaj kodni blok izpisuje poruku upozorenje u slučaj bekasi i prekida rad programa. Međutim, funkcija koja obrađuje grešku može se promeniti. To se radi funkcijom **ErrorBlock**.

```
ErrorBlock({|e| MojaObrada(e)})
```

Kada se izvrši ove naredbe, obrađa greške se prenumerava na funkciju **Greška**. Svaka pojava greške u vreme izvršavanja programa može se zavrhava funkcijom Greška. Ova funkcija ima ne razpolagaju tačnome promenljive kreiranog objekta koje opisuju kakva se greška dogodila. Na primer interne promenljiva **description** sadrži tekstualno opis greške **genCode** kod greške i sl Funkcija **Greška** može ispisati poruku o nastalj greška i savršiti rad ili pokušati so nastavljam programa Evo zaluh važnih internih promenljivih objekta za obradu grešaka.

canRetry	indikator koji označava da b je moguće nastaviti program i pozad greška.
description	opis greška
filename	nazv datoteke kod čije obrade je nastala greška
genCode	kod greška
operation	operacija
osCode	DOS kod greška

OET OBJEKTI

Ranije je bilu reči o tome kako se prave maske se unos korišćenjem **SAY..GET** naredbi U otvari unas podstake je rezlozvan preko tzv **GET** objekata, a **SAY GET** naredbe se pretpreosmaju i odgovarajuća operacije se objektima Ideja je slična u ska slučaja. svaki **GET** objekat sadrži sve informacije o unesu jedna promenljive Na primer interna promenljiva *raw* sadrži rad n kome se vrednost učitave ood kalenu **unmas**